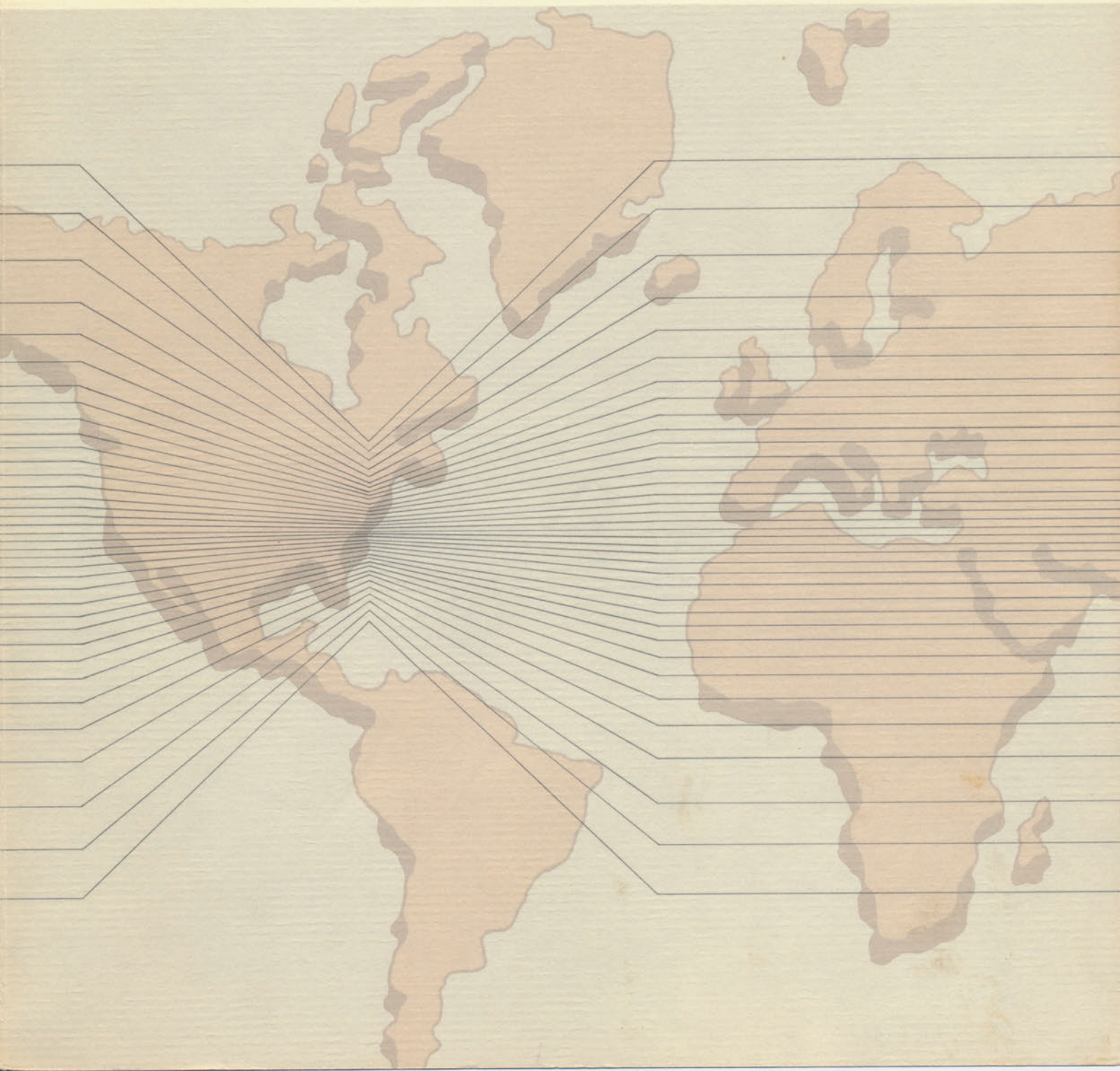
I. P. Sharp Associates

MABRA
Users' Guide



MABRA

Users' Guide

Acknowledgements

MABRA was designed and implemented in early 1978 by Hugh Hyndman and Raj Chauhan. Steve Halasz is currently responsible for its continuing support and development. This manual was written and illustrated by Steve Halasz and Katy Gormley, and revised by Wendy Sexsmith.

Printed in Canada.
Publication Code: 0002 8310 E2
ISBN 0 86493 083 6

CONTENTS

		Page
1	INTRODUCTION	1
	What is MABRA?	1
	Who Should Use MABRA?	3
	Features of MABRA	4
	The Structure of This Manual	5
2	GLIMPSE OF MABRA	6
	Setting Up a MABRA System	6
	Working With the Records in a MABRA System	8
	Reporting the Data in a MABRA System	12
3	MABRA BASICS	14
	Getting Started	14
	Different Date Formats	16
	Using MABRA Systems Other Than Your Own	16
	Listing MABRA Systems	17
	Entering *DATA MODE*	18
4	DISPLAYING DATA AND PRODUCING REPORTS	19
	Listing Records	19
	Batch Printing Large Listings	24
	Analyzing and Reporting Relationships Among Selected Fields	25
	Producing Specialized Summaries and Listings	33
	Linking Data From Other MABRA Systems	40
	Specifying and Printing Standard Forms	53
5	MAINTAINING MABRA SYSTEMS	65
	Adding Records	65
	Deleting Records	70
	Changing Records	71
	Listing Update Requests	73
	Withdrawing Update Requests	75
	Updating a MABRA System	76
	Removing Update Requests	77
	Copying Records From Other MABRA Systems	78
6	SETTING UP AND MANAGING MABRA SYSTEMS	80
	Creating a MABRA System	80
	Adding Fields	81
	Deleting Fields	88
	Changing Fields	89
	Listing Fields	93
	Code Description Tables	95
	Defining Access to a MABRA System	97
	Locking a MABRA System	105
	Resizing a MABRA System	106
	Reblocking the Master File of a MABRA System	106
	Renaming a MABRA System	107
	Displaying the Status of a MABRA System	108
	Erasing a MABRA System	109

Chapter 1

INTRODUCTION

What is MABRA?

MABRA is I.P. Sharp's online data management system.

So much of the data collected and maintained by businesses today is routine and very structured. For example, consider your company's personnel data. For each employee, you keep their name, address, date of birth, and salary on file. For each of your customers and distributors, you keep track of their name, address, name of contact person, and which products they purchase from you. Your inventory data, sales and expenditure data, budget data, and shipping and receiving data also falls into this category of routine, structured information. The data that must be maintained in many of these cases is so well-defined that you may have pre-printed forms designed to assist you in capturing and conveniently storing it. The following form, which is designed to record the data for a company employee, may be similar to a form used by your organization.

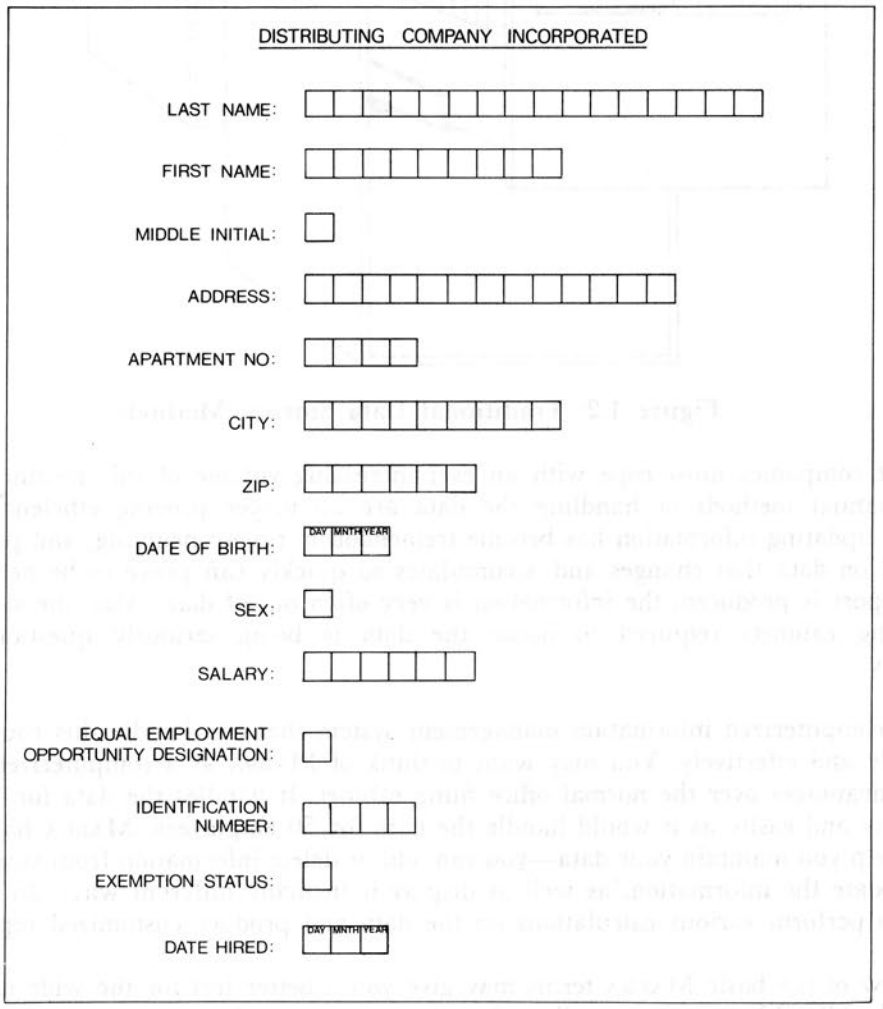


Figure 1.1: Employee Data Form

Typically, these forms would be stored in one of the many filing cabinets lining the walls of your office.

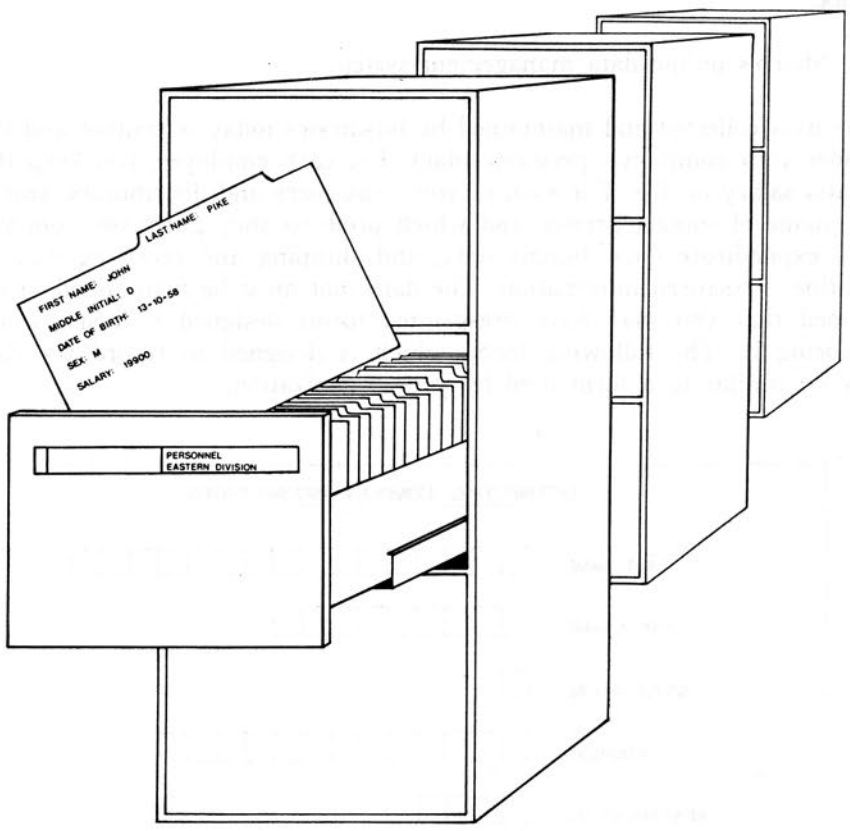


Figure 1.2: Traditional Data Storage Methods

Because most companies must cope with an ever-increasing volume of this routine information, the traditional manual methods of handling the data are no longer proving efficient or cost-effective. Locating and updating information has become tremendously time-consuming, and producing different reports based on data that changes and accumulates so quickly can prove to be near impossible—by the time a report is produced, the information is very often out of date. Also, the storage cost for the miles of filing cabinets required to house the data is being seriously questioned by company administrators.

MABRA is a computerized information management system that can handle this routine, well-defined data efficiently and effectively. You may want to think of MABRA as a computerized filing cabinet—with many advantages over the normal office filing cabinet. It handles the data for 10,000 employees as conveniently and easily as it would handle the data for 50 employees. MABRA has all the facilities required to help you maintain your data—you can add or delete information from your MABRA system, change or update the information, as well as display it in many different ways. In addition, MABRA allows you to perform various calculations on the data and produce customized reports.

Defining a few of the basic MABRA terms may give you a better feel for the wide range of data that MABRA can handle. MABRA is normally referred to as a **record management system**. If you kept all your personnel data in a MABRA system, all the items of information pertaining to one employee (e.g., name, date of birth, salary) would constitute one **record**. If your company employed 5,814 people, you would probably have 5,814 records in your MABRA system.

Each of the items of information which you maintain for one employee (e.g., name, address, home telephone number) is called a **field**. Therefore, a MABRA system can be thought of as having two dimensions—one dimension being the records of information and the second dimension being the fields. The following diagram illustrates these two dimensions of a MABRA system.

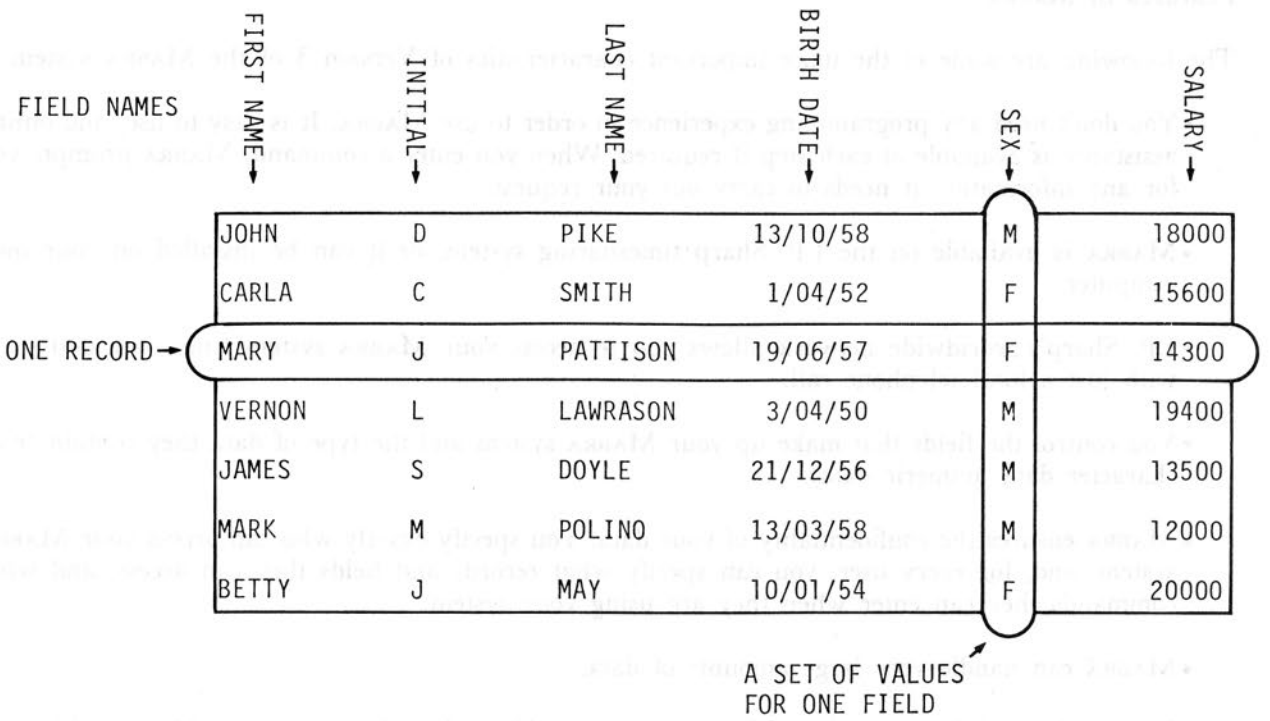


Figure 1.3: MABRA's Organization of Data

One MABRA record can contain at least 200 different fields, and at least 150,000 records can be stored in one MABRA system (the exact limitations vary depending on workspace size). Each user account number can support up to 99 distinct MABRA systems.

Who Should Use MABRA?

MABRA has been designed to be used by non-programmers. By learning a few basic commands, you can store, list, or change the information in your system. Associated with each command is a series of prompts that asks you, in plain English, to enter any details or data that MABRA requires to complete your request. If you respond to a prompt in a way that is unacceptable to MABRA, you are informed by means of a question mark followed by an appropriate message. The prompt is then repeated, allowing you to enter a correct response. If you are unsure of what to type at any point during a MABRA session, you can enter *HELP* and an explanation of the acceptable responses will be displayed.

As your knowledge of MABRA grows, there are a number of features available that allow you to save time and use MABRA more efficiently. For example, if you enter a particular set of commands and responses every week to produce a certain report, you can save time and effort by saving that set of commands and responses and assigning to it a unique name. Then, each week you would simply enter the unique name. These stored sets of MABRA instructions are called **stored answer-aheads**.

For those who know or wish to learn certain programming procedures, *MABRAUTIL*, a collection of MABRA utilities, allows you to use MABRA as a high-level, non-procedural record management system.

Thus MABRA can be used by people at various levels of sophistication.

Features of MABRA

The following are some of the more important characteristics of Version 3 of the MABRA system:

- You don't need any programming experience in order to use MABRA. It is easy to use, and online assistance is available at each step if required. When you enter a command, MABRA prompts you for any information it needs to carry out your request.
- MABRA is available on the I.P. Sharp timesharing system, or it can be installed on your own computer.
- I.P. Sharp's worldwide network allows you to access your MABRA system from over 600 cities with just a local telephone call.
- You control the fields that make up your MABRA system and the type of data they contain (e.g., character data, numeric data).
- MABRA ensures the confidentiality of your data. You specify exactly who can access your MABRA system, and, for every user, you can specify what records and fields they can access, and what commands they can enter when they are using your system.
- MABRA can handle very large amounts of data.
- You can locate information in a MABRA system quickly and easily using any field or combination of fields as the "search key".
- You can have updates to a MABRA system processed immediately (to save time), or they can be processed in a low priority batch mode (to save money).
- Data that is added to a MABRA system can be validated in different ways. For example, you can have MABRA check that the data is the correct type or that it falls within a certain range.
- You can save sets of instructions that need to be entered frequently under a unique name—eliminating constant re-typing.
- You can have detailed analyses performed on a set of records using MABRA's cross-tabulation facilities.
- You can have reports printed at your terminal or routed to a highspeed printer and delivered to you.
- You can link two or more MABRA systems so that reports can include data from more than one system.
- Your MABRA system can include pseudo fields, which are fields that are not stored but are produced using the values from other fields.

- For larger MABRA systems, you can have fields that are searched frequently inverted for more efficient searching.

- Fields can be coded to save storage space as well as time and effort when data is being entered. The meaning associated with the various codes can be stored and used in the production of reports.

- A utility workspace is provided for manipulating a system's raw data so that those with APL programming experience can write customized report and updating functions.

- MABRA is compatible with other I.P. Sharp products such as MAGIC and XTABS.

The Structure of This Manual

This manual should be read by anyone who will be creating or maintaining MABRA systems, storing or updating data in a MABRA system, or producing reports from the stored data.

Chapter 2 gives a quick demonstration of MABRA at work, showing how a system is created and maintained, and how a report is generated. The fundamental concepts of MABRA are discussed in Chapter 3, along with a few of the basic commands. Chapter 4 gives a description of all the commands used to display the data from a MABRA system and produce informative reports. Chapter 5 explains how data is put into a MABRA system—how records are added, changed, and deleted, and how systems are updated. In Chapter 6, the commands used in setting up a MABRA system are illustrated. A complete discussion of field types and formats is included. This chapter also describes the commands required by system stewards and designers for managing and reorganizing MABRA systems.

Cost saving features and suggestions for using MABRA more efficiently are highlighted throughout the manual.

If you are a programmer and are interested in MABRA's design specifications and facilities for custom programming, you should also read the MABRA *Programmer's Guide*.

If you have any questions about MABRA that this manual or the *Programmer's Guide* do not answer, you should call your I.P. Sharp representative, or send a MAILBOX message to the group *MABQ*.

Chapter 2

GLIMPSE OF MABRA

This chapter gives a brief overview of how MABRA is used to perform some of the basic record management functions. In the first section, the MABRA program is started up, a MABRA system is created, and fields for that system are defined. In the second section, records are added, changed, and displayed, and the MABRA system is updated. In the final section, a short report is produced. Detailed explanations of all the commands illustrated are given in the chapters that follow.

A sample personnel system has been set up and is used to demonstrate the MABRA features and commands throughout this manual. If you wish, you can sign on to the I.P. Sharp timesharing system and duplicate the examples found in this manual.

Note: To access the systems used in this manual, you must type *SETCREATOR* 86 immediately after loading the MABRA workspace.

When the MABRA commands, associated prompts, messages, and responses are illustrated, the entries that you must type are printed in shaded areas. For example:

```
* PRINT FIELDS: LASTNAME,FIRSTNAME
```

means that the prompt * *PRINT FIELDS* was typed by MABRA, and you responded with *LASTNAME,FIRSTNAME*.

Setting Up a MABRA System

This section demonstrates the commands required to initiate the MABRA program (*START*), create and enter a MABRA system (*CREATE*, *ENTER*), and define fields for that system (*AF*).

```
      )LOAD 86 MABRA
SAVED  14.36.56 03/07/83
COPYRIGHT I.P. SHARP ASSOCIATES LIMITED 1981
      START
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

The first thing you must do in order to use MABRA is load the workspace that contains the MABRA program. Remember that after typing an entry at the terminal, you must press the RETURN key (carriage return) to send your entry to the computer.

After the MABRA workspace is loaded and a message has been displayed indicating when the MABRA program was last saved, you must type *START* to begin the MABRA program. MABRA replies that Version 3.1 of the MABRA program has been initiated and that you are in *SYSTEM MODE*.

There are two modes in MABRA—*SYSTEM MODE* and *DATA MODE*. When you begin a MABRA session, you are in *SYSTEM MODE* and you can create new MABRA systems, and erase, rename, enter, or list the names of existing MABRA systems. You are in *DATA MODE* when you enter a particular MABRA system and you can then organize, add, change, or report information for that system.

MABRA's prompt for a command in *SYSTEM MODE* is a colon (:).

```
:CREATE
* CREATE SYSTEM
* ENTER SYSTEM NUMBER: 1
* ENTER NAME FOR SYSTEM 01: PERSONNEL
* ENTER EXPECTED MAXIMUM NUMBER OF RECORDS: 100
* SYSTEM 01: PERSONNEL
* CREATE? (Y/N): Y
* SYSTEM CREATED.
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

This conversation with MABRA created a system numbered "1" or "01" and named *PERSONNEL*. It is expected to contain a maximum of 100 records (although the *RS* command described in Chapter 6 will allow you to increase this limit if necessary).

```
:ENTER
* SYSTEM NUMBER: 1
* DATA MODE, SYSTEM IS 01: PERSONNEL
01:
```

You have created a MABRA system, but it is empty. Before you can put any data into the system, you must *ENTER* the system and define or add the fields. When you *ENTER* a MABRA system, you move from *SYSTEM MODE* to *DATA MODE*. The prompt for a command in *DATA MODE* is the system number followed by a colon (e.g., 01:).

```
01:AF
* ADD FIELD
* ENTER FIELD NAME: LASTNAME
* TYPE? (CHAR/INTE/REAL/BINA/DATE): CHAR
* NUMBER OF ELEMENTS IN FIELD: 20
* USE DEFAULT OUTPUT FORMAT? (Y/N): Y
* OK TO ADD? (Y/N): Y
* UPDATING FILES.....
* FIELD ADDED.
* ENTER FIELD NAME:
```

To add a field to a MABRA system, you enter the *AF* command in response to the *DATA MODE* prompt. MABRA then asks you to supply a number of details pertaining to the field. After you finish supplying all the required information, you are prompted to add more fields. This pattern of returning to the first prompt is typical of what takes place with many MABRA commands. Once you enter a command, a series of prompts is repeated until you enter *STOP* in response to the first prompt. This is a helpful feature as you often want to add more than one field when you enter *AF*, as in this particular demonstration. If you no longer wish to add fields, you would enter *STOP* and the *DATA MODE* prompt (e.g., 01:) would be displayed once again. You could then enter another MABRA command. The following dialogue with MABRA adds fields named *FIRSTNAME*, *BIRTHDATE*, and *SALARY*.

```
* ENTER FIELD NAME: FIRSTNAME
* TYPE? (CHAR/INTE/REAL/BINA/DATE): CHAR
* NUMBER OF ELEMENTS IN FIELD: 15
* USE DEFAULT OUTPUT FORMAT? (Y/N): Y
* OK TO ADD? (Y/N): Y
* UPDATING FILES.....
* FIELD ADDED.
```

```
* ENTER FIELD NAME: BIRTHDATE
* TYPE? (CHAR/INTE/REAL/BINA/DATE): DATE
* USE DEFAULT OUTPUT FORMAT? (Y/N): Y
* OK TO ADD? (Y/N): Y
* UPDATING FILES.....
* FIELD ADDED.
* ENTER FIELD NAME: SALARY
* TYPE? (CHAR/INTE/REAL/BINA/DATE): INTE
* NUMBER OF ELEMENTS IN FIELD: 1
* USE DEFAULT OUTPUT FORMAT? (Y/N): Y
* OK TO ADD? (Y/N): Y
* UPDATING FILES.....
* FIELD ADDED.
* ENTER FIELD NAME: STOP
01:
```

Working With the Records in a MABRA System

This section illustrates the commands that allow you to add records to your MABRA system (*AR*), change them so they are kept up to date (*CD*), and display them (*LR*). Before you can type any of these commands, you must be in *DATA MODE* for a particular system.

```
01:AR
* ADD RECORD(S)

[1]
LASTNAME  :PIKE
FIRSTNAME :JOHN
BIRTHDATE :131058
SALARY    :18000

[2]
LASTNAME  :
```

When you enter the *AR* command, MABRA proceeds to prompt you for values for each of the fields that you have defined for the system you have entered. You are prompted for each field in the order that the fields were originally added, unless you have previously defined other field orders with the *FO* command (described in Chapter 5). After you have entered a value for each of the existing fields, you are prompted again for the first field and you can continue to add another record.

```
[2]
LASTNAME  :PATTISON
FIRSTNAME :MARY
BIRTHDATE :061957
? INVALID FIELD TYPE, RE-ENTER
BIRTHDATE :190657
SALARY    :14300

[3]
LASTNAME  :LAWRAWON
FIRSTNAME :VERNON
BIRTHDATE :030450
SALARY    :→LASTNAME
```

```
LASTNAME  :LAWRASON
FIRSTNAME :→SALARY
SALARY    :19400

[4]
LASTNAME  :
```

As you can see from the example above, when you enter a value that is invalid for a particular field, MABRA informs you and re-prompts for that field.

In the event that you discover an error as you are adding a record, the "go to" (→) feature of MABRA allows you to go back to a particular field in the record that is being added and re-enter it. You are then prompted for the field that normally follows the field which you backed up to. If you already entered a correct value for that field, you can use "go to" (→) to skip forward to a particular field.

```
[4]
LASTNAME  :POLONIO;MARK;130358;12000

[5]
LASTNAME  :MCDUFF;MICHAEL; ;14234;MAY;BETTY;100154
SALARY    :20000

[7]
LASTNAME  :
```

Once you know what prompts you will be getting, you can reduce the time you spend communicating with the computer by "answering ahead". As you can see from the example above, you do this by typing the entries one after another on the same line, separated by semicolons (;). This is a fast and economical way to use MABRA.

If you do not have values for certain fields, you can type a blank space, and the field for that record will be left empty.

```
[7]
LASTNAME  :SMITH;M;121245;18777

[8]
LASTNAME  :DAVIS;C;→

[9]
LASTNAME  :STOP
* UPDATE REQUEST NO. 1 FILED.
01:
```

When you use → without following it with a field name, you skip over any fields you have not entered values for and go on to the next record. Any fields that are skipped over are left empty.

You enter *STOP* to indicate that, for the time being, you no longer wish to add any more records. MABRA then files a request to update the system and assigns an identification number to the update request. This request will be to add the eight records as entered above to the MABRA system.

It is important to be aware that MABRA doesn't immediately perform additions, changes, or deletions to a system's records. Rather, each request to add, change, or delete records is filed or stored. This allows you to review update requests before the changes or additions are actually made.

To actually have one or more update requests processed, you must use the *UP* command.

```
01:UP
* INITIATE UPDATE TASK
* 1 REQUEST PENDING UPDATE.
* RUN AS N-TASK OR B-TASK? (N/B): N
* N-TASK NO.2448
01:EXIT
      )OFF
```

When you enter the *UP* command, MABRA tells you how many requests are pending update and then asks you whether the update process should be run as an N-task or a B-task. N-tasks and B-tasks are two kinds of jobs that MABRA can run to update a system, and they are discussed in more detail in Chapter 5. Briefly, an N-task runs immediately so the update task is processed quickly, while a B-task runs when the computer isn't as busy and is therefore more economical.

When you initiate an update task either via an N-task or a B-task, it is processed separately from your work at the terminal. Thus you don't have to wait for the task to complete. Instead, you can go ahead and do other work, or you can exit from the MABRA program and sign off the system. The N-task or B-task will complete on its own.

The first thing you wish to do when you begin your next MABRA session is display the eight records which you added in the first session.

```
      )LOAD 86 MABRA
SAVED  14.36.56 03/07/83
COPYRIGHT I.P. SHARP ASSOCIATES LIMITED 1981
      START
* MABRA VERSION 3.1
* SYSTEM MODE
:ENTER
* SYSTEM NUMBER: 1
* LAST UPDATE:  1/08/83 18:55:29; CPU USED: 6
* NUMBER OF COMPLETED UPDATES: 1
* DATA MODE, SYSTEM IS 01: PERSONNEL
01:LR
* LIST RECORDS
* CONSTRAINTS: ALL
     8 FOUND.
* PRINT FIELDS: FIRSTNAME,LASTNAME,αBIRTHDATE,SALARY
* SORT FIELDS: LASTNAME
* ALIGN PAPER, TYPE SPACE, CR TO START.

FIRSTNAME       LASTNAME        αBIRTHDATE      SALARY

C               DAVIS                                0
VERNON          LAWRASON                33       19400
BETTY           MAY                     29       20000
MICHAEL         MCDUFF                           14234
MARY            PATTISON                26       14300
JOHN            PIKE                    24       18000
MARK            POLONIO                 25       12000
M               SMITH                   37       18777
* CONSTRAINTS: STOP
01:
```

You use the *LR* command to list one or more records from a MABRA system. After entering this command, the first thing you are prompted for is a constraint. Constraints are a means of selecting the records that you wish to work with. The simplest constraint is *ALL* which selects all the records in the system. You can be more specific by entering constraints such as *LASTNAME=PIKE* or *SALARY*>18000. Constraints are discussed in detail in Chapter 4. After you have entered a constraint, MABRA responds by telling you how many records in your system meet your constraint. You are then asked to specify the fields from each record that you wish to have printed (and they will be printed in the same order as they are specified), as well as the fields(s) that you wish to have your records sorted by.

In addition to printing a field's value, MABRA can often print a variation of the value depending on the type of data which the field contains. If you type α in front of a field that contains date values, as in the example above, MABRA will calculate the number of years between the date stored in that field and the current date. Thus, by specifying α*BIRTHDATE* as a print field, MABRA calculates and displays each employee's present age.

Next, each employee has just been given a salary increase, so your system must be updated to reflect the new salary figures.

```
01:CD
* CHANGE DIFFERENT VALUES
* CHANGE FIELDS: SALARY
* MORE? (Y/N): N
* PRINT FIELDS: LASTNAME,FIRSTNAME,SALARY
* SORT FIELDS: LASTNAME
* CONSTRAINTS: ALL
     8 FOUND.
[1] DAVIS                ;C                ;          0
  SALARY: 13736

[2] LAWRASON             ;VERNON           ;      19400
  SALARY: 20050

[3] MAY                  ;BETTY            ;      20000
  SALARY: 21770

[4] MCDUFF               ;MICHAEL          ;      14234
  SALARY: 15755

[5] PATTISON             ;MARY             ;      14300
  SALARY: 15650

[6] PIKE                 ;JOHN             ;      18000
  SALARY: 18788
```

```
[7] POLONIO              ;MARK          ;     12000
  SALARY: 14210
[8] SMITH                ;M             ;     18777
  SALARY: 19770
* UPDATE REQUEST NO. 2 FILED.
* CONSTRAINTS: STOP
* CHANGE FIELDS: STOP
01:
```

The *CD* command allows you to change one or more fields in each record identified by a constraint to a different value. The fields specified for *PRINT FIELDS* are displayed for each record before the prompt for a new field value.

Reporting the Data in a MABRA System

Once you have stored data in a MABRA system, you can quickly and easily report and analyze that data using MABRA's reporting commands. The command *LR*, which allows you to list one or more fields from specified records, has already been demonstrated. Other reporting commands, however, allow you to cross-tabulate data from a MABRA system and produce summaries or listings with one or more levels of subtotalling. There are even commands which can be used to produce mailing labels and form letters. If you need to produce reports using data from more than one MABRA system, there is also a command that links fields from other systems.

GR (Generate Report) is the only report command that will be demonstrated in this chapter. It lets you print summary or detailed listings of numeric field values, so you can produce a wide range of analytic, financial, and accounting reports, such as sales summaries by district or salesman, budget analyses, financial statements, and inventories. This next example uses *GR* to summarize the headcount and salaries of all employees.

```
01:GR
* GENERATE REPORT.
* CONSTRAINTS: ALL
     8 FOUND.
* PRINT FIELDS: LASTNAME,FIRSTNAME,SALARY,ΔCOUNT:NUMBER
* SORT FIELDS: LASTNAME
* TOTAL FIELDS: ΔCOUNT,SALARY
* PRINT RECORD DETAILS? (Y/N): Y
* REPORT WIDTH IS 52 CHARACTERS.
* ANY SPECIAL OPTIONS? (Y/N): N
* ALIGN PAPER, TYPE SPACE, CR TO START.
```

```
DATE:  1/08/83                                        PAGE 1
                        01: PERSONNEL
                    CONSTRAINTS: ALL

LASTNAME        FIRSTNAME                SALARY NUMBER

DAVIS           C                             0      1
LAWRASON        VERNON                    19400      1
MAY             BETTY                     20000      1
MCDUFF          MICHAEL                   14234      1
PATTISON        MARY                      14300      1
PIKE            JOHN                      18000      1
POLONIO         MARK                      12000      1
SMITH           M                         18777      1
                                         ---------- ------
                                          116711      8

* CONSTRAINTS: STOP
01:
```

Note that the updated salary values are not reflected in this report as the *CD* update request has not yet been processed.

GR controls paging automatically if the report requires more than one sheet of paper to print. A page heading, which includes the date, page number, system name and number, and constraints, is also generated automatically.

Δ*COUNT* is a pseudo field created by MABRA that contains the value 1 for every record, and it is used to count records as in this example. Pseudo fields are explained in detail in Chapter 6.

Chapter 3

MABRA BASICS

Getting Started

In order to use MABRA, you must first load the public workspace in library 86 that contains the MABRA program. This is done by entering:

```
      )LOAD 86 MABRA
SAVED  14.36.56 03/07/83
COPYRIGHT I.P. SHARP ASSOCIATES LIMITED 1981
```

Note: Throughout this manual, the entries that you or any MABRA user would type are shown in the shaded areas. In this way, you can easily distinguish the user commands and responses from MABRA's prompts and messages.

Remember that after typing any command or entry at the terminal, you must press the RETURN key (carriage return) to send your entry to the computer. Once the MABRA workspace has been loaded, a message is displayed indicating when the MABRA program was last saved.

Next, you must initiate the MABRA program by typing:

```
      START
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

The *START* command initiates the latest version of the MABRA program and puts you in *SYSTEM MODE*. Appropriate messages to this effect are displayed.

There are basically two modes in MABRA—*SYSTEM MODE* and *DATA MODE*. You are in *SYSTEM MODE* immediately after initiating the MABRA program with *START*. In *SYSTEM MODE* you can create new MABRA systems, list the names of existing MABRA systems, and erase, rename, or enter MABRA systems. The commands used to perform some of these system functions will be covered in this chapter, while the others will be described in Chapter 6. You move from *SYSTEM MODE* to *DATA MODE* when you enter a particular MABRA system. You are then able to specify how data is to be organized for that system; add, change, or delete data; and produce informative reports from the stored data.

MABRA's prompt for a command while you are in *SYSTEM MODE* is a colon (:), while the prompt for a command in *DATA MODE* is the number of the system you have entered, followed by a colon (e.g., 01:). Besides these two main prompts, any message that ends with a colon is a prompt from MABRA for more information.

If you are ever unsure of what to enter at any point during a MABRA session, you can always type *HELP* and a list of the valid responses will be displayed. If you require further explanation, you can type *HELP* a second time. For example:

```
* SYSTEM MODE
:HELP
ENTER ONE OF: CREATE;ERASE ;RENAME;NEWPAS;LOCK  ;ENTER ;LIST  ;HELP;STOP
:HELP
```

```
ENTER ONE OF:
        CREATE  - TO CREATE A NEW MABRA SYSTEM
        ERASE   - TO ERASE AN EXISTING MABRA SYSTEM
        RENAME  - TO RENAME AN EXISTING MABRA SYSTEM
        NEWPASS - TO INPUT CREATOR'S SIGN-ON PASSWORD
        LOCK    - TO LOCK A SYSTEM WITH MAGIC NUMBERS (P)
        ENTER   - TO ENTER (OR ACCESS) AN EXISTING MABRA SYSTEM
        LIST    - TO LIST ALL EXISTING MABRA SYSTEMS
        STOP    - TO EXIT BACK TO APL IMMEDIATE EXECUTION MODE
:
```

If you ever enter a command or response that is incorrect or unacceptable, MABRA responds with a question mark and a message that explains the error. You are then reprompted for your command or response.

Another useful command that can be entered at any point during a MABRA session is *STOP*. *STOP* performs a slightly different function depending on when it is entered. For example:

- If *STOP* is entered in response to the *SYSTEM MODE* command prompt (:), you exit from MABRA and return to APL immediate execution mode (i.e., you are back to the point you were at just after you signed on to the I.P. Sharp timesharing system).

- If *STOP* is entered in response to the *DATA MODE* command prompt, you leave *DATA MODE* and return to *SYSTEM MODE*.

- You can also enter *STOP* in response to a prompt from any one of the *SYSTEM MODE* or *DATA MODE* commands. When a command has a corresponding series of one or more prompts that is repeated over and over, you must type *STOP* to indicate that you wish to terminate the command that is issuing the prompts. For example, when you enter the *ADD RECORDS* command (*AR*), a certain set of prompts is repeated so you can add as many records as you wish. When you have no more records to add, you type *STOP*. The *DATA MODE* command prompt is then displayed and you can enter a different command if you choose.

  Sometimes when *STOP* is entered in response to a prompt issued by a command, the command is not terminated but you are returned to an earlier prompt. Thus *STOP* is used to indicate that you have finished entering a certain part of the information required by the command function. If you actually want to terminate the command, you would have to type *STOP* again.

  STOP can also be used in certain circumstances when you discover that an error has been made, and you want to terminate the command and start again, or "back up" to a certain prompt so you can specify the correct information.

The use of *STOP* will be demonstrated frequently throughout this manual, and you will become familiar with the function it serves at different times during a MABRA session.

EXIT is another command that can be entered at any point during a MABRA session and in response to any MABRA prompt. It allows you to leave the MABRA program at any time and return to APL immediate execution mode. If you enter *EXIT* in response to a prompt issued by a MABRA command, that command will be aborted.

If you are using any *DATA MODE* command, you can terminate the command at any point and return immediately to the *DATA MODE* command prompt (e.g., `01:`) with the *BACK* command. The use of *BACK* will also be illustrated in this manual.

Different Date Formats

Normally date values are entered or displayed via MABRA in the format **ddmmyy** (the first two digits representing a day of the month, the second two digits representing a month, and the last two digits representing a year). However, you can choose an alternate date format by selecting the *USDATES* state setting. When this state is set, all dates are displayed and must be entered in the form **mmddyy**. The *USDATES* state must be selected before the MABRA program is initiated with the *START* command. For example:

```
      )LOAD 86 MABRA
SAVED  14.36.56 03/07/83
COPYRIGHT I.P. SHARP ASSOCIATES LIMITED 1981
      USDATES
WAS NOUSDATES
      START
* MABRA VERSION 3.1
* DATES ARE IN FORM MM/DD/YY
* SYSTEM MODE
:
```

As indicated above, the normal or default state is *NOUSDATES*.

Using MABRA Systems Other Than Your Own

Every MABRA system has a number and a name, both of which are assigned by the user when the system is created. In addition to these identifiers, MABRA assigns to each new system a value for *CREATOR*. The value assigned is simply the account number of the user who created the system. MABRA classifies all existing systems according to their *CREATOR*. For example, all systems belonging to account 1234 are grouped together, while all those owned by account 5678 are part of another group. Figure 3.1 illustrates this organization of MABRA systems.

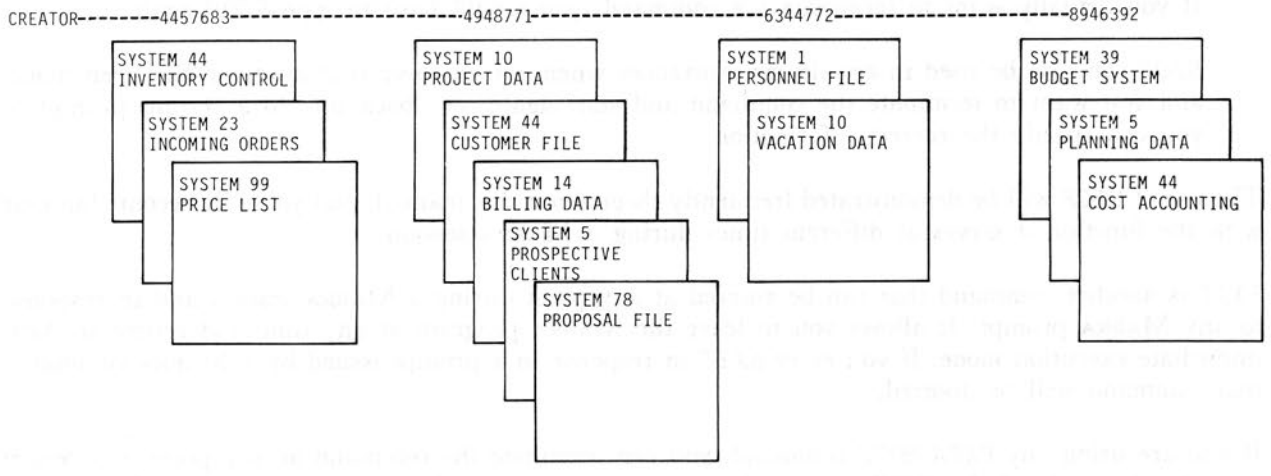


Figure 3.1: Organization of MABRA Systems

From the diagram above, you can see that there can be more than one MABRA system with the same number. However, within any one group of systems, each system number is unique. This is why, during a MABRA session, you can only work with one group of systems at a time, and MABRA assumes when the workspace is first loaded that you want to work with the group for which you are the *CREATOR*. Thus *CREATOR* is initially set to your account number.

If you want to use a system that belongs to another group, you must change the setting of *CREATOR* to the account number of that group's owner. The *SETCREATOR* command, used as follows, allows you to change the setting of *CREATOR*. You must be aware, however, that *SETCREATOR* can only be entered *before* the MABRA program has been initiated with *START*.

SETCREATOR **nnn**

where **nnn** is the account number that owns the system you want to use. Assuming the user who created the system you wish to use has given you access to it, you can proceed to initiate MABRA, enter the system, and perform any of the functions you have been given permission to execute. Defining and setting access to MABRA systems is discussed in Chapter 6.

In this manual, it is assumed that your account is 1234567 and that you are the creator or owner of the systems used in the examples.

Listing MABRA Systems

As discussed in the previous section, the setting of *CREATOR* determines the set or group of systems that you can work with during the session. If you wish to list the system numbers and corresponding names of the MABRA systems in this group, you use the *LIST* command.

```
* SYSTEM MODE
:LIST
* LIST SYSTEM(S)
* ALIGN PAPER, TYPE SPACE, CR TO START.
```

Whenever MABRA is about to print a report or summary of any kind, a prompt to align the paper is first displayed for the benefit of those using hardcopy terminals. This provides you with an opportunity to reposition the paper before the report is printed. When you are ready to have the report printed, you type a blank space followed by a carriage return. Then, once the report has been printed, MABRA waits for you to again type a space followed by a carriage return before any further prompts are displayed. You therefore have another opportunity to reposition the paper.

This feature allows you to keep the actual report separate from the prompts and responses required to produce it.

The following is a sample of the type of report produced by the *LIST* command.

```
MABRA SYSTEMS ON USER: 1234567  DATE:  1/08/83

NO    NAME

01    PERSONNEL
10    EDUCATION
20    COURSES
```

Entering *DATA MODE*

SYSTEM MODE, as mentioned, is the state in which MABRA systems are created and erased, and the state in which the names of systems are listed and changed. But, to develop and display the content of a MABRA system, you must be in *DATA MODE*. You move from *SYSTEM MODE* to *DATA MODE* by executing the *ENTER* command as follows:

```
* MABRA VERSION 3.1
* SYSTEM MODE
:ENTER
* SYSTEM NUMBER: 1
* LAST UPDATE:  7/08/83 14:06:23; CPU USED: 12
* NUMBER OF COMPLETED UPDATES: 4
* DATA MODE, SYSTEM IS 01: PERSONNEL
01:
```

You are immediately prompted for the number of the system you want to enter. When you respond with a valid system number, MABRA first tells you when that system was last updated (and how many CPU units were used by the last update task). Then you are reminded of how many completed update requests are currently being stored.

Storing completed updates does increase costs, so they should be removed periodically using the *RC* command (see Chapter 5).

Next, MABRA confirms that you are now in *DATA MODE* and displays the number and name of the system you are using. MABRA is then ready to accept any *DATA MODE* command. The prompt for a command in *DATA MODE* is the number of the system you are working with, followed by a colon.

In the example above, system 1 is entered.

The entire time you are in *DATA MODE* you are working with one particular system, and the command prompt (e.g., `01:`) serves as a reminder of which system it is. If you complete your work with one system and want to work with another system, you must first return to *SYSTEM MODE* by entering *STOP* in response to the *DATA MODE* command prompt. For example:

```
01:STOP
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

You then execute the *ENTER* command as above, and, when prompted, you specify the number of the next system you wish to work with.

Chapter 4
DISPLAYING DATA AND PRODUCING REPORTS

Record management involves two main activities—putting data in and getting data out. This chapter discusses the features of MABRA and the commands related to the latter activity.

MABRA includes a number of very powerful report generating commands that allow you not only to display the data stored in a MABRA system, but more importantly, to transform the raw data into useful information. There are commands that let you list records or specific fields from one or more records. Other reporting commands cross-tabulate data from a MABRA system, or produce summaries or listings with one or more levels of subtotalling. For mailing labels or form letters, MABRA has commands for setting up and printing form or template-type reports. And, when you need to produce reports using data stored in different MABRA systems, there is a command that links fields from other systems.

We will continue to use the personnel system to demonstrate MABRA's reporting commands.

To use any reporting command, you must be in *DATA MODE* which means you must have *ENTER*ed a particular MABRA system.

Listing Records

The simplest type of report is a listing of records, and you can produce this type of report at your terminal using the *LR* command. When you enter *LR*, you are prompted for three things: *CONSTRAINTS*, *PRINT FIELDS*, and *SORT FIELDS*. These are all prompts that are common to other reporting commands as well.

A **constraint** is the way you tell MABRA which records you are interested in. You specify a constraint by combining a field name, a relationship, and a value. For example:

SALARY>25000

where *SALARY* is the field name, > (greater than) is the relationship, and 25000 is the value. A relationship can be any one of the following:

Symbol	Meaning
<	Find values that are **less than** a particular value
≤	Find values that are **less than or equal to** a particular value
=	Find values that are **equal to** a particular value
≠	Find values that are **not equal to** a particular value
≥	Find values that are **greater than or equal to** a particular value
>	Find values that are **greater than** a particular value
⊃	Find values that **contain** a particular value
∈	Find values that are **members of** a list of values

You can combine two or more conditions in a single constraint by connecting them with ∧ (and) or ∨ (or). For example:

SALARY>25000∧*SEX=F*

This constraint would locate the records of all female employees whose salaries are greater than

$25,000. When the ∧ (and) symbol is used, MABRA selects a record only if it meets both the conditions that are being combined. It therefore tends to *restrict* your selection of records. If the ∨ (or) symbol is used, MABRA selects a record if it meets either one or both of the conditions that are being combined. This has the effect of *expanding* your selection of records.

All constraints are evaluated from left to right, unless you use parentheses to alter the order. Thus the constraint:

```
SEX=F∧SALARY>20000∨HIREDATE>010178
```

would select the records of all *female* employees who are paid more than $20,000 annually, or the record of *any* employee who has been hired since January 1, 1978. Adding parentheses to the same constraint as follows:

```
SEX=F∧(SALARY>20000∨HIREDATE>010178)
```

would select the records of all *female* employees whose salaries are more than $20,000 *or* who were hired after January 1, 1978.

MABRA also allows you to perform a string search with a constraint. For example:

```
LASTNAME⊃MAN
```

would locate all records with a *LASTNAME* value that contains the three characters *MAN* together in any location.

After you enter a valid constraint, MABRA responds by informing you of how many records in the current system meet your constraint.

When you are prompted for *PRINT FIELDS*, you specify the names of the fields whose values you want to have displayed. Field names must be separated by commas. The order in which you specify the fields determines the order in which they are displayed across the page. You can enter *ALL* if you want all fields included in the listing.

A field name can be preceded by a ~ (*tilde*) which means "not". This symbol is normally used in association with *ALL*. If you wanted *ALL* fields to be printed with the exception of one or two, it would normally be easier to enter *ALL* followed by the names of the fields to be excluded, each preceded by ~ (e.g., *ALL,~SALARY*), than it would be to specify each print field explicitly.

LR uses the field names as column headings unless you give an alternate heading. When you define your print fields, you can specify an alternate column heading for one or more fields by following the appropriate field name(s) with a colon and the new heading. If you want to specify a multi-line heading, you separate each line of the heading with a colon. For example:

```
* PRINT FIELDS: IDNUMBER:EMPLOYEE:NUMBER,LASTNAME,SALARY
```

would define a two-line heading for *IDNUMBER*.

If you are including a coded field in your listing (see Chapter 6 for a description of coded fields) and you want the appropriate code descriptions printed in place of the field values, you enclose the field name in parentheses. For example:

```
* PRINT FIELDS: LASTNAME,BIRTHDATE,(EEO)
```

MABRA can sort the records identified by a constraint so they are printed in a particular order. For example, you often want to have employee records displayed alphabetically or in descending order according to salary. When you are asked for *SORT FIELDS*, you can specify any field or combination of fields. Sort fields do not have to be print fields as well. If you specify more than one sort field, the fields specified first (the left-most fields) are the major sort fields. You can sort in two directions—**ascending** or **descending**. Ascending is the standard (or default) order, but you can request a descending sort for a field by preceding the field name with a minus sign (-). When you do not want the records displayed in any particular order, you enter *NONE* in response to *SORT FIELDS*.

LR is used in the following example to list the *LASTNAME*, *SEX*, and *SALARY* of all employees whose salaries are greater than $45,000. The records are sorted in descending order by salary.

```
01:LR
* LIST RECORDS
* CONSTRAINTS: SALARY>45000
    7 FOUND.
* PRINT FIELDS: LASTNAME,SEX,SALARY
* SORT FIELDS: -SALARY
* ALIGN PAPER, TYPE SPACE, CR TO START.

LASTNAME              SEX     SALARY

BAYES                 M    84,333.00
BATES                 M    49,200.00
BLACK                 F    48,900.00
BOND                  F    48,600.00
BERNARD               F    47,200.00
COLE                  F    45,500.00
COLLINS               M    45,100.00
```

After a listing is printed, MABRA again prompts for *CONSTRAINTS*, allowing you to define another set of records to be displayed.

Besides printing a field's value, MABRA can, in certain cases, print a variation or a derivative of the value. For example, if you type α in front of a field that contains *date* values, MABRA will calculate the number of years between the date stored in that field and the current date. Therefore, when you specify α*BIRTHDATE* in a constraint or as a print field, MABRA calculates each employee's present age.

```
* CONSTRAINTS: PREVIOUS^SEX=F
    4 FOUND.
* PRINT FIELDS: LASTNAME,αBIRTHDATE,SALARY
* SORT FIELDS: BIRTHDATE
* ALIGN PAPER, TYPE SPACE, CR TO START.

LASTNAME              αBIRTHDATE      SALARY

BLACK                         58  48,900.00
COLE                          58  45,500.00
BERNARD                       48  47,200.00
BOND                          23  48,600.00
```

PREVIOUS is a special constraint used to select all the records from the previous constraint. The previous constraint can be from another command or the same command. *PREVIOUS* can either stand on its own, or it can be combined with other

conditions as in the example above. The ability to use the result of the previous constraint is a very economical feature of MABRA. You save the cost and computer time that would be spent by MABRA having to perform the same search again.

Another special constraint is *ALL* which selects all the records currently stored in the system.

You can precede any constraint with ~ (tilde) and, as in the case of print fields, it means "not". Thus the constraint *~PREVIOUS* would select all records not identified by the previous constraint. *~EEO=0* would select all records whose Equal Employment Opportunity (*EEO*) field does not equal "*0*". Sometimes it is easier to specify what you don't want than it is to specify what you do want. In such cases, ~ is very useful.

MABRA automatically assigns a unique record number to each record added to a MABRA system. This number represents the location of the record in the master file. The quickest and most efficient way for MABRA to locate a particular record is by its unique record number. Thus, if you want to display a certain record or group of records and you know their record numbers, you should specify these numbers in a constraint. Record numbers must be separated by blanks if more than one is entered.

```
* CONSTRAINTS: 23 176 29
     3 FOUND.
* PRINT FIELDS: LASTNAME,(EEO),SALARY
* SORT FIELDS: -SALARY
* ALIGN PAPER, TYPE SPACE, CR TO START.

LASTNAME              (EEO)                SALARY

MILLER                WHITE             35,886.00
LONG                  HISPANIC          20,647.00
NASH                  ORIENTAL          20,198.00
```

If you enter the record number of a deleted record or a record that you do not have access to, MABRA behaves as if you did not enter that number.

Because locating records by their record number is so efficient, it is useful to have a listing of records that includes each record's unique number. You might, for example, request a listing of records that includes the record number field and a field like *LASTNAME* so record numbers can easily be looked up. The name of the field that contains the record number is Δ.

You should be aware that if the master file for the system is reblocked (see Chapter 6) and deleted records are purged, record numbers will change.

```
* CONSTRAINTS: ALL
   196 FOUND.
* PRINT FIELDS: Δ,LASTNAME
* SORT FIELDS: LASTNAME
* ALIGN PAPER, TYPE SPACE, CR TO START.

  Δ LASTNAME

147 ACKER
181 ACKER
 81 ALBERT
```

```
 68 ALEXANDER
125 ALEXANDER
138 ALEXANDER
 12 ALLEN
 31 ALLEN
133 ALLEN
100 ANDERSON
121 ANDERSON
118 ANDREWS
 77 ANTHONY
128 ANTHONY
162 ANTHONY
 34 BACON
119 BACON
153 BAILEY

.
.
.

* CONSTRAINTS: STOP
01:
```

Δ is a session pseudo field that MABRA adds automatically to every system. Pseudo fields are described in detail in Chapter 6.

If you enter *ALL* in response to the prompt for *PRINT FIELDS*, MABRA will print values for all fields that have been added to the system by users. The fields that are added by MABRA (Δ, Δ*COUNT*) will only be included if they are specified in addition to *ALL* (e.g., *ALL*,Δ).

When you use character fields in a constraint, it is important to realize that if you use a value in the constraint that has fewer elements than the number of elements in the field, MABRA finds all records with field values that *begin* with the value you specify. For example, if you enter the constraint *LASTNAME*=*BALL*, MABRA will select all records where the value stored in the *LASTNAME* field starts with the letters "*BALL*" (e.g., *BALL*, *BALLARD*, *BALLANTINE*). *LASTNAME* is a field that contains 20 elements or characters. If the name stored in *LASTNAME* is fewer than 20 characters, the rest of the field is filled with blank characters. Therefore, if you wanted MABRA to find the records of all employees whose name was *BALL* (and not *BALLARD* or *BALLANTINE*), you would type a blank space after "*BALL*" when you are entering your constraint.

When you are using date fields in a constraint, you can specify a complete date including a day, month, and year (e.g., *HIREDATE*>010178), or a month and year (e.g., *BIRTHDATE*=0180), or simply a year.

```
BIRTHDATE=44
```

would locate the records of all employees who were born in 1944, while:

```
HIREDATE=180
```

would select the records of all employees hired in January 1980.

When you want MABRA to search only certain elements of a field, you must specify the elements to be searched in the following way:

```
* CONSTRAINTS: IDNUMBER[2 3]=15
```

This constraint would result in MABRA finding the record of any employee who has 1 as the second element of his *IDNUMBER* and 5 as the third element. You use [] (brackets) in a constraint when you want MABRA to search only certain elements of a field that contains more than one element.

Batch Printing Large Listings

When you want to print a large number of records, sorting and listing them at your terminal can be costly and certainly time consuming. The economical alternative is to have the listing printed on the nearest I.P. Sharp highspeed printer and delivered to you. The *BP* (Batch Print) command is used for this purpose.

The output generated by the *BP* command is identical to the output from *LR*, except that a heading prints at the top of each page. The heading includes the creator, number, and name of the system that the records are from, as well as the constraints specified to produce the listing.

When you enter *BP*, you are prompted first for *CONSTRAINTS*, *PRINT FIELDS*, and *SORT FIELDS*. You respond to these prompts in the same way described previously for the *LR* command. After answering these prompts, you are asked to supply print specifications and delivery instructions.

Highspeed print specifications (e.g., *NOERASE*, *DO2*) are explained in detail in the I.P. Sharp publication, *SATN-8*, which is available from your local I.P. Sharp office. Delivery instructions must include your name and complete address, as well as any special shipping instructions. When you finish entering the delivery instructions (taking as many lines as you need), you type *STOP* on a line by itself.

Next, you are asked to specify whether the print request should be run as an N-task or a B-task.

N-tasks and B-tasks are two methods that MABRA has of getting a job done. An N-task runs immediately, so it is the appropriate choice when you are in a hurry to receive the record listing. A B-task runs when the demand for the computer resources is low (after 23:00 U.T.C.), so it is usually the more economical choice.

Once you submit a B-task for a particular system, you cannot process an N-task for that same system until the B-task has completed. When updates are discussed in Chapter 5, N-tasks and B-tasks are discussed again. For additional information on N-tasks and B-tasks, see the I.P. Sharp publication, *Batch Tasks in SHARP APL*.

Whether you choose an N-task or a B-task, your print request is taken care of separately from the work that is being done during your terminal session. You don't have to wait for the job to complete, so you can go ahead and do other work, or you can leave MABRA and sign off the system. The N-task or B-task that is taking care of your print request will complete on its own.

When you have confirmed that the request should be submitted, MABRA responds with the number of the print file created to process your print request and the B-task or N-task number of your *BP* highspeed print job.

The following example illustrates the use of *BP*:

```
01:BP
* BATCH PRINT
* CONSTRAINTS: ALL
   196 FOUND.
```

```
* PRINT FIELDS: Δ,LASTNAME,FIRSTNAME,INITIAL,IDNUMBER
* SORT FIELDS: LASTNAME,FIRSTNAME,INITIAL
* PRINT SPECS: DO2
* ENTER DELIVERY INSTRUCTIONS. TYPE 'STOP' TO END INSTRUCTIONS.
DISTRIBUTING COMPANY, INC.
EASTERN DIVISION PERSONNEL
847 EAST DRIVE
ROCHESTER, NY 14932
ATTN:  L. WHITTAKER
STOP
* RUN AS N-TASK OR B-TASK? (N/B): B
* SUBMIT REQUEST? (Y/N): Y
* PRINT FILE IS: P1023195647
* B-TASK REQ.NO. 492766
* CONSTRAINTS: STOP
01:
```

When the *BP* job completes, details regarding the status of the job are displayed the next time you enter that same MABRA system or use the *NS* command (described in Chapter 5). These details would include the N-task or B-task number, the SORTREQ number (if sorting was required for the job), the HSPRINT request number, and the time of job completion.

Analyzing and Reporting Relationships Among Selected Fields

MABRA shows its real power and utility by its ability to analyze your data in various ways and produce informative reports needed for effective decision-making. The *AN* command can analyze your data in the following ways:

- *AN* can count the number of times a certain value appears in a particular field. The report resulting from this analysis shows the **frequency** with which a certain field value occurs.
- *AN* can add numeric field values with the resulting report showing the **totals** for each category you specify.
- *AN* can calculate **average values** for numeric fields.
- MABRA can combine two or more of the above analysis functions in a single report.

The table below lists the options available to you with *AN*.

Option	Type of Analysis	Number of Reports
F	Counts field values	1
T	Totals numeric fields	1
A	Averages numeric fields	1
FT	Counts field values; totals numeric fields	2
FA	Counts field values; averages numeric fields	2

TA	Totals numeric fields; averages numeric fields	2
FTA	Counts field values; totals numeric fields; averages numeric fields	3

Besides offering these basic report options, *AN* can enhance reports in the following ways:

- *AN* can calculate what percentage of an entity each field value represents.
- *AN* can calculate the grand total of a field's values. Depending on the option you choose, this would represent the grand total of a frequency count, numeric totals, or an overall average.

When you devise a cross-tabulation analysis, an important consideration is that the fields you intend to analyze should have a *limited number of values*. For example, suppose you wanted to see how salaries are distributed in your company according to the employees' sex. Your analysis would compare the *SALARY* and *SEX* fields. *SEX* has only two possible values (*F* and *M*), which is quite manageable. *SALARY*, however, could have as many different values as there are employees. Comparing such a large number of values would be a long and ineffectual process, so you have to limit the number of values in order to make a stronger and more concise comparison. You do this by putting the field values into **ranges**. *AN* allows you to specify the boundaries defining the ranges that the field values will fall into. Once you have a limited number of possibilities for all the fields to be compared, you can devise a report that is short, easy to interpret, and informative.

A simple cross-tabulation example will serve to illustrate the steps involved in the production of such a report.

```
01:AN
* CROSSTABULATION ANALYSIS
* OPTION? (F/T/A/FT/FA/FTA/TA): F
* CONSTRAINTS: ALL
  196 FOUND.
* FIELD NAME SCHEME: SEX
* USE RANGES? (Y/N): N
* PRINT PERCENTAGES? (Y/N): N
* DO ANALYSIS? (Y/N): Y
* CPU USED: 9
* ALIGN PAPER, TYPE SPACE, CR TO START.

                          01: PERSONNEL
* FREQUENCY ANALYSIS. CONSTRAINTS: ALL. FIELDS: SEX.

SEX=F :        60
   =M :       136
TOTAL :       196

GRAND TOTAL : 196
```

To produce a cross-tabulation report, you must first specify one of the *AN* options, and, in this example, the *F* (frequency count) option was selected. *AN* then asks for a constraint to find out what records

in the system are to be involved in the analysis. This particular analysis involved *all* records in the system.

The prompt *FIELD NAME SCHEME* is asking you to specify what fields are to be analyzed for the report. Field names must be separated by commas if more than one is entered. Field derivatives can also be entered in response to this prompt. For the above report, only the field *SEX* was analyzed. There are only two possible values for the field *SEX* (*F* and *M*) so there is no need to define ranges. Some of the examples that follow show how ranges are specified. This example also did not include percentages. The last prompt asks you to confirm that the responses you have given are correct and that you want MABRA to proceed with the analysis. If you have made any errors or wish to change one of your responses, you would enter *N* at this point and start defining your cross-tabulation once again.

MABRA uses a separate batch task to perform cross-tabulations, and then reports the amount of computer time, in CPU's (computer processing units), taken to perform the analysis.

The report itself includes the system's number and name, followed by a descriptive heading which includes the type of analysis performed, the constraints specified, and the names of the fields involved in the analysis. The body of the report shows each value for the *SEX* field and associated with each value is a number that indicates how many times that value occurs within the group of records defined by the constraint.

Because this is a simple frequency count of one field, the field's total and the grand total are the same. Also, because the constraint *ALL* was used, the grand total equals the total number of records in the system.

After a cross-tabulation report is printed (and you have typed a space followed by a carriage return), *AN* again prompts for a report option. You would type *STOP* in response to this prompt when you have finished producing cross-tabulation reports.

The next report is also a frequency analysis, but this time two fields, *SEX* and *SALARY*, are involved. In order to include *SALARY* in an analysis, a limited number of ranges that every salary value will fall into must be defined. It doesn't make sense to count the occurrences of each different salary value, as there could be as many different salary amounts as there are records identified by the constraint.

```
* OPTION? (F/T/A/FT/FA/FTA/TA): F
* CONSTRAINTS: ALL
  196 FOUND.
* FIELD NAME SCHEME: SEX,SALARY
* USE RANGES? (Y/N): Y
* FIELD NAMES ARE:
  1. SEX
  2. SALARY

* ENTER NUMBER OF FIELD NAME: 2
* SALARY

:12000
:20000
:28000
:36000
:45000
:STOP
* ENTER NUMBER OF FIELD NAME: STOP
```

```
* PRINT PERCENTAGES? (Y/N): Y
* BASED ON WHICH TOTAL? (R/C/G): R
* DO ANALYSIS? (Y/N): Y
* CPU USED: 3
* ALIGN PAPER, TYPE SPACE, CR TO START.

                              01: PERSONNEL
* FREQUENCY ANALYSIS. CONSTRAINTS: ALL. FIELDS: SEX,SALARY.

          C1      C2      C3      C4      C5      C6   TOTAL
       ------------------------------------------------------
SEX=F |    5      35       7       3       4       6      60
      |  8.33   58.33   11.67    5.00    6.67   10.00  100.00

   =M |   30      48      16      13      13      16     136
      |22.06   35.29   11.76    9.56    9.56   11.76  100.00

TOTAL |   35      83      23      16      17      22     196


GRAND TOTAL : 196


  C1 :              SALARY<12000
  C2 : 12000     ≤SALARY<20000
  C3 : 20000     ≤SALARY<28000
  C4 : 28000     ≤SALARY<36000
  C5 : 36000     ≤SALARY<45000
  C6 : 45000     ≤SALARY
```

When you indicate that you want to use ranges in your analysis, MABRA lists the field names that you specified when prompted for *FIELD NAME SCHEME*. A number is printed along with each field name, and you are prompted to enter the number associated with the field you want to define ranges for. MABRA then prints the name of the field selected and prompts for each range point with a colon (:). In this example, the boundaries for dividing up the different salary values are specified, starting with the lowest salary range point. When you have finished defining range points, you type *STOP*. MABRA then gives you the opportunity to define range points for another field. If ranges have been defined for all fields that require them, you enter *STOP* to proceed to the next prompt.

The table produced in this example has two rows and six columns. There is one row for each different value of *SEX* that is found in the records identified by the constraint. The columns are labelled *C1*, *C2*, *C3*, etc., and the meaning associated with each label is given in the legend printed below the table. In this case, each column represents a different range for *SALARY*.

For each different combination of *SEX* and *SALARY* range, the report gives a frequency count which represents the number of records having that combination of field values. For example, a frequency count of 48 appears in row 2 and column 2, meaning there were 48 records found with a *SEX* value of *M* as well as a *SALARY* value in the range $12,000 to $19,999. In addition to the individual frequency counts, totals for each column and row are automatically calculated and printed.

In this particular report, percentages are also included along with the frequency counts and totals. When you are setting up your analysis and specify that percentages are to be printed, you are asked whether the percentages are to be based on row totals, column totals, or the grand total. In this

example, the rows represent different values for *SEX*. So by selecting percentages based on row totals, the report shows how each different *SEX* category is broken down by *SALARY*.

The third cross-tabulation is a frequency, totalling, and averaging analysis. Three different reports are produced in this case, one for each option. This time the fields involved in the analysis are *EEO* and *SALARY*.

```
* OPTION? (F/T/A/FT/FA/FTA/TA): FTA
* CONSTRAINTS: ALL
  196 FOUND.
* FIELD NAME SCHEME: EEO,SALARY
* USE RANGES? (Y/N): Y
* FIELD NAMES ARE:
  1. EEO
  2. SALARY

* ENTER NUMBER OF FIELD NAME: 2
* SALARY

:12000
:20000
:28000
:36000
:45000
:STOP
* ENTER NUMBER OF FIELD NAME: STOP
* FIELD TO BE TOTALLED/AVERAGED: SALARY
* PRINT PERCENTAGES? (Y/N): Y
* BASED ON WHICH TOTAL? (R/C/G): R
* DO ANALYSIS? (Y/N): Y
* CPU USED: 3
* ALIGN PAPER, TYPE SPACE, CR TO START.
```

Again, ranges are defined for *SALARY* and the columns of each table represent the different *SALARY* ranges. Each table contains six rows, one for each value of *EEO*.

When you choose the totalling (*T*) or averaging (*A*) options, you are asked to specify which field is to be totalled and/or averaged. In this example, *SALARY* is the field that is totalled and averaged. Field derivatives can also be entered.

```
                                01: PERSONNEL
* FREQUENCY ANALYSIS. CONSTRAINTS: ALL. FIELDS: EEO,SALARY.

          C1     C2     C3     C4     C5     C6   TOTAL
      ------------------------------------------------
EEO=A|                    2                          2
    |                100.00                     100.00
   =B|      7     14      5      4      2      2     34
    |  20.59  41.18  14.71  11.76   5.88   5.88 100.00
   =H|     11     12      2                         25
    |  44.00  48.00   8.00                     100.00
   =M|      2     14      9     11     14      5     55
    |   3.64  25.45  16.36  20.00  25.45   9.09 100.00
   =O|      2      4      1                          7
    |  28.57  57.14  14.29                     100.00
   =W|     13     39      4      1      1     15     73
    |  17.81  53.42   5.48   1.37   1.37  20.55 100.00
TOTAL|     35     83     23     16     17     22    196

GRAND TOTAL : 196


  C1 :           SALARY<12000
  C2 : 12000    ≤SALARY<20000
  C3 : 20000    ≤SALARY<28000
  C4 : 28000    ≤SALARY<36000
  C5 : 36000    ≤SALARY<45000
  C6 : 45000    ≤SALARY
```

The first report produced gives the frequency counts for the different values of *EEO* in combination with the different *SALARY* ranges. A percentage based on the row total is printed below each frequency count.

```
                                01: PERSONNEL
* TOTALS. CONSTRAINTS: ALL. FIELDS: EEO,SALARY. TOTALLING: SALARY

          C1       C2       C3       C4       C5       C6    TOTAL
      ------------------------------------------------------------
EEO=A|                    41907                             41907
    |                   100.00                            100.00
   =B|    67539   213454   121480   130900    77700    90600   701673
    |     9.63    30.42    17.31    18.66    11.07    12.91   100.00
   =H|   102735   197691    41079                             341505
    |    30.08    57.89    12.03                            100.00
   =M|    22700   227719   224200   348700   568400   278233  1669952
    |     1.36    13.64    13.43    20.88    34.04    16.66   100.00
   =O|    20535    56736    20198                              97469
    |    21.07    58.21    20.72                            100.00
   =W|   121723   616969    85125    35886    40387   675000  1575090
    |     7.73    39.17     5.40     2.28     2.56    42.85   100.00
TOTAL|   335232  1312569   533989   515486   686487  1043833  4427596

GRAND TOTAL : 4427596


  C1 :           SALARY<12000
  C2 : 12000    ≤SALARY<20000
  C3 : 20000    ≤SALARY<28000
  C4 : 28000    ≤SALARY<36000
  C5 : 36000    ≤SALARY<45000
  C6 : 45000    ≤SALARY
```

In the second report, the *SALARY* values for the records with each *EEO/SALARY* combination are totalled. For example, the entry in row 5 and column 2 means that the total salary for all oriental employees whose salaries fall in the range $12,000 to $19,999 is $56,736. The percentage printed immediately below means that this total of $56,736 represents 58.21 percent of the total salary for *all* oriental employees.

```
                                01: PERSONNEL
* AVERAGES. CONSTRAINTS: ALL. FIELDS: EEO,SALARY. AVERAGING: SALARY

            C1      C2      C3      C4      C5      C6     TOTAL
      ------------------------------------------------------------
EEO=A|                    20954                            20954
    |                     100.00                           100.00

   =B|    9648   15247   24296   32725   38850   45300     20637
    |   46.75   73.88  117.73  158.57  188.25  219.50    100.00

   =H|    9340   16474   20540                             13660
    |   68.37  120.60  150.36                             100.00

   =M|   11350   16266   24911   31700   40600   55647     30363
    |   37.38   53.57   82.04  104.40  133.72  183.27    100.00

   =O|   10268   14184   20198                             13924
    |   73.74  101.87  145.06                             100.00

   =W|    9363   15820   21281   35886   40387   45000     21577
    |   43.39   73.32   98.63  166.32  187.18  208.56    100.00

TOTAL|    9578   15814   23217   32218   40382   47447     22590


GRAND TOTAL : 22590


  C1 :              SALARY<12000
  C2 : 12000     ≤SALARY<20000
  C3 : 20000     ≤SALARY<28000
  C4 : 28000     ≤SALARY<36000
  C5 : 36000     ≤SALARY<45000
  C6 : 45000     ≤SALARY
```

The third report shows the average salary in each salary range for employees in each of the *EEO* categories. The table is actually the result of dividing the first report (frequency count) into the second report (total salaries). In this averaging report, the figures appearing in the column and row labelled *TOTAL* are not totals in this instance, but rather averages for the entire row or column. For example,

the figure 13660 in the *TOTAL* column for *EEO=H* is the total combined salary for all hispanic employees (from the second report) divided by the total number of hispanic employees (from the first report). The *GRAND TOTAL* in this report is the average salary of *all* employees.

The last cross-tabulation example is a frequency analysis involving three fields—*EXEMPT*, *EEO*, and *SEX*. All of these fields have a limited number of possible values so ranges are not required. As we have seen in previous examples where two fields are specified for the *FIELD NAME SCHEME*, the values of the last field specified make up the columns of the table, while the values for the first field form the rows. When three fields are involved in the analysis, the report is made up of more than one table, as the two dimensions of a table are no longer sufficient.

In this example, where *EXEMPT*, *EEO*, and *SEX* are specified for the *FIELD NAME SCHEME*, the different values of *SEX* make up the columns of each table, while the values of *EEO* make up the rows. Then there is a separate table for each different value of *EXEMPT*. Since *EXEMPT* is a binary field, it can only have two different values, 0 and 1. Thus the report is made up of two tables. One table shows the frequency of each *EEO/SEX* combination when *EXEMPT* equals 0; the other, when *EXEMPT* equals 1.

```
* OPTION? (F/T/A/FT/FA/FTA/TA): F
* CONSTRAINTS: ALL
  196 FOUND.
* FIELD NAME SCHEME: EXEMPT,EEO,SEX
* USE RANGES? (Y/N): N
* PRINT PERCENTAGES? (Y/N): N
* DO ANALYSIS? (Y/N): Y
* CPU USED: 5
* ALIGN PAPER, TYPE SPACE, CR TO START.

                              01: PERSONNEL
* FREQUENCY ANALYSIS. CONSTRAINTS: ALL. FIELDS: EXEMPT,EEO,SEX.


EXEMPT=0

            C1      C2     TOTAL
        ----------------------
EEO=A|       1       1         2
   =B|       6      18        24
   =H|       2      14        16
   =M|      14      24        38
   =0|       4       2         6
   =W|      12      39        51
TOTAL|      39      98       137


------
```

```
EXEMPT=1

          C1      C2     TOTAL
       ----------------------
EEO=A|
   =B|       4       6        10
   =H|       3       6         9
   =M|       3      14        17
   =O|       1                  1
   =W|      10      12        22
TOTAL|      21      38        59
------

GRAND TOTAL : 196
  C1 : SEX=F
  C2 :    =M

* OPTION? (F/T/A/FT/FA/FTA/TA): STOP
01:
```

Producing Specialized Summaries and Listings

The *GR* (Generate Report) command allows you to produce a wide range of reports, from simple record listings to complex reports that summarize and analyze your data in various ways. What this command offers over other report commands is a great amount of control over the organization and format of your report. For example, you can print listings of records similar to the listings produced with *LR* or *BP*, but with *GR* you have more control over paging, page size, page title, and spacing. You can also produce summaries or analyses by specifying that one or more fields are to be subtotalled by sorted categories.

When you use the *GR* command, you are prompted first for *CONSTRAINTS*, as you are with the other report commands discussed so far. Next, you are asked to supply *PRINT FIELDS*, *SORT FIELDS*, and *TOTAL FIELDS*.

Fields are specified at the *PRINT FIELDS* prompt for any of the following reasons:

- If you want to include a field in a report and it will not be specified as a *SORT FIELD* or a *TOTAL FIELD* (*SORT FIELDS* and *TOTAL FIELDS* are included in a report automatically so there is no need to specify them again as *PRINT FIELDS*).
- To control the order in which fields are displayed from left to right across the page.
- If a field is a coded field (see Chapter 6), and you want to print the field's code description in the report (rather than the code which may not be as meaningful).
- If you want to specify an alternate column heading for a field (unless you specify otherwise, a field's name is used as the column heading).
- If you want to print variations of field values. For example, if you want date values converted to the appropriate number of years.

The examples in this section will illustrate these different reasons for entering *PRINT FIELDS*.

When you specify *SORT FIELDS*, you are indicating how you want your report organized. The

SORT FIELDS you select also control the subtotalling. You must specify the *SORT FIELDS* in order, with the major sort fields given first.

TOTAL FIELDS are simply the numeric fields you want MABRA to calculate totals for.

After entering *TOTAL FIELDS*, you are asked if you want to have record details printed in the report. You reply *Y* or *N* depending on whether you want field values from each record included in your report or only subtotals and/or totals.

At this point MABRA can determine how wide the report will be, and a message containing this information is displayed.

Finally, you are asked if you wish to select any special options. *GR* has a number of special options that allow you to control the printing of your report. They will be discussed later in this section.

At the beginning of this chapter, a pseudo field called Δ was discussed. This field is created automatically by MABRA for every record, and it contains a record's unique record number. In the following example, another pseudo field is introduced, and it is also created automatically by MABRA for every record. *ΔCOUNT* contains the value 1 for every record, and it is used for totalling records.

```
01:GR
* GENERATE REPORT.
* CONSTRAINTS: ALL
   196 FOUND.
* PRINT FIELDS: NONE
* SORT FIELDS: SEX,EEO
* TOTAL FIELDS: ΔCOUNT
* PRINT RECORD DETAILS? (Y/N): N
* REPORT WIDTH IS 30 CHARACTERS.
* ANY SPECIAL OPTIONS? (Y/N): N
* ALIGN PAPER, TYPE SPACE, CR TO START.

DATE:  9/08/83          PAGE 1
         01: PERSONNEL
       CONSTRAINTS: ALL

SEX EEO ΔCOUNT

F   A         1
F   B        10
F   H         5
F   M        17
F   O         5
F   W        22
M   A         1
M   B        24
M   H        20
M   M        38
M   O         2
M   W        51
        ------
           196
```

In this example, *NONE* was entered in response to *PRINT FIELDS* as all the fields to be included in the report were either *SORT FIELDS* or *TOTAL FIELDS*.

MABRA automatically prints the current date and a page number at the top of each page, and unless you specify differently with a special option, MABRA automatically formats and centres a report title. The default title shown in the above report contains the number and name of the currently *ENTER*ed MABRA system and the constraints used to select the records included in the report. *GR* uses the field names as column headings unless alternate headings are specified, as do *LR* and *BP*.

The report shows each different combination of values for the two *SORT FIELDS*, *SEX* and *EEO*, followed by the total of *ΔCOUNT* for each combination. This report is similar to a cross-tabulation analysis produced with *AN*.

The next example illustrates how *GR* is used to calculate and print subtotals along with a listing of individual records.

In this case, a number of fields are entered in response to *PRINT FIELDS*. *FIRSTNAME* and *INITIAL* are included because they are neither *SORT FIELDS* nor *TOTAL FIELDS*. *LASTNAME* and *SEX* are entered because the fields are to be printed in a certain order, starting with *LASTNAME*. *ΔCOUNT* is included in the list in order to give an alternate column heading. This is done, as with *LR* and *BP*, by following the field name with a colon (:) and the new column heading. If the new heading is to consist of more than one line, you separate each line of the heading by a colon. For example:

```
* PRINT FIELDS: ΔCOUNT:NUMBER:OF:EMPLOYEES
```

would result in a three-line column heading.

When you are specifying *SORT FIELDS* and you use a / (slash) as a field name separator instead of a comma, MABRA will calculate subtotals for each different value of the *SORT FIELD(S)* to the left of the slash. In this example, a subtotal is printed for each different value of *SEX*.

```
* CONSTRAINTS: EEO=0
    7 FOUND.
* PRINT FIELDS: LASTNAME,FIRSTNAME,INITIAL,SEX,ΔCOUNT:NUMBER
* SORT FIELDS: SEX/LASTNAME
* TOTAL FIELDS: ΔCOUNT,SALARY
* PRINT RECORD DETAILS? (Y/N): Y
* REPORT WIDTH IS 69 CHARACTERS.
* ANY SPECIAL OPTIONS? (Y/N): N
* ALIGN PAPER, TYPE SPACE, CR TO START.
```

```
DATE: 9/08/83                                          PAGE 1
                        01: PERSONNEL
                    CONSTRAINTS: EEO=O

LASTNAME        FIRSTNAME       INITIAL SEX NUMBER      SALARY

DAVIS           CATHERINE       P       F       1    13,736.00
HARRIS          SUSAN           G       F       1    18,275.00
MARTIN          SUSAN           B       F       1     9,404.00
NASH            CARLA           L       F       1    20,198.00
ROBERTS         LOUISE          G       F       1    12,106.00
                                                ------ -----------
                                        F       5    73,719.00

ELLIOTT         SAMUEL          M       M       1    12,619.00
HALL            FRED            O       M       1    11,131.00
                                                ------ -----------
                                        M       2    23,750.00
                                                ====== ===========
                                                7    97,469.00

* CONSTRAINTS: STOP
01:
```

This report includes the requested field values from the 7 records identified by the constraint *EEO=O*. The records have been sorted by *SEX* and, within each of the two groups resulting from that sort, the records are ordered according to *LASTNAME*. Because the slash was used when the *SORT FIELDS* were entered, the values for *ΔCOUNT* and *SALARY* are subtotalled for each different value of *SEX*. When you specify *TOTAL FIELDS* with *GR*, you get a grand total printed under a double totalling line at the end of the report, in addition to any subtotalling you request by using a slash when specifying *SORT FIELDS*.

When you are generating reports, you may find you need to include data that is computed from other field values. For example, you might want to print each employee's monthly salary in a report, when only the annual salary is stored in each employee record. To produce the monthly figure required, a simple calculation would have to be performed on the annual salary. To include computed data in a report, you must first create a field or fields to store the new data. There is a special type of field, called a pseudo field, that stores data based on calculations. A complete discussion of pseudo fields and how they are created is found in Chapter 6.

Pseudo fields are also useful for transforming existing field values into more convenient forms or data types. To generate the next report, two pseudo fields, *EEOBINARY* and *SEXEEO*, are created to convert the values for *EEO* and *SEX* to a binary form. *EEOBINARY* is a six-element binary field with one element containing the value 1, while the other elements contain the value 0. Which element contains the value 1 depends on the value contained in the record's *EEO* field. *SEXEEO* is based on *EEOBINARY*, but also considers the value for *SEX*. It is a twelve-element binary field, again with one element containing the value 1 and the remaining elements containing the value 0. Which element contains the value 1 this time depends on the values contained in the fields *EEO and SEX*. If the record describes a male employee, the value 1 will appear in the first six elements, and if it corresponds to a female employee, the 1 will be found in the last six elements. *EEOBINARY* is defined as a **session** pseudo field, which means that any values it contains will disappear and the field will cease to exist when you exit from the MABRA program. Because *SEXEEO* is based on a session pseudo field (*EEOBINARY*), it must also be defined as a session pseudo field.

The *AP* command is used to add a pseudo field. When this command is entered, you are prompted first for a field name and then for details such as the type of data to be stored, the number of elements which the data will consist of, the output format for the data, and the expression used to compute the data. Finally, you must indicate whether or not the field is to be retained after the MABRA session has ended. All of *AP*'s prompts and the required responses are fully explained in Chapter 6.

```
01:AP
* ADD PSEUDO FIELD
* ENTER FIELD NAME: EEOBINARY
* TYPE? (CHAR/INTE/REAL/BINA/DATE): BINA
```

```
* NUMBER OF ELEMENTS IN FIELD: 6
* USE DEFAULT OUTPUT FORMAT? (Y/N): Y
* ENTER EXPRESSION DEFINING FIELD
:(,EEO)∘.='WBOMHA'
* RETAIN THIS FIELD? (Y/N): N
* OK TO ADD FIELD? (Y/N): Y
* PSEUDO FIELD ADDED.
* ENTER FIELD NAME: SEXEEO
* TYPE? (CHAR/INTE/REAL/BINA/DATE):BINA
* NUMBER OF ELEMENTS IN FIELD: 12
* USE DEFAULT OUTPUT FORMAT? (Y/N): Y
* ENTER EXPRESSION DEFINING FIELD
:(6×SEX='F')ΦEEOBINARY,(ρEEOBINARY)ρ0
* RETAIN THIS FIELD? (Y/N): N
* OK TO ADD FIELD? (Y/N): Y
* PSEUDO FIELD ADDED.
* ENTER FIELD NAME: STOP
01:
```

Now *GR* is used to total the new binary pseudo field, *SEXEEO*, for a frequency analysis. Some different ways of specifying *PRINT FIELDS* are shown in this example. When a *PRINT FIELD* is enclosed in parentheses, as *EXEMPT* is in this case, the code descriptions are printed in the report in place of the field values. Thus the two descriptions, *NONEXEMPT* and *EXEMPT*, appear in place of the two different values for *EXEMPT*, 0 and 1. Typing a colon after a *PRINT FIELD* is the manner in which a new column heading is specified. However, when the colon is followed by a blank space (as with *EXEMPT*), a blank column heading (no column heading) will appear for that field. On the other hand, if you type nothing after the colon (as with *SEXEEO*), you are then prompted separately, line by line, to enter a column heading for the field. The first line of the column heading for *SEXEEO* is designed to show which elements of the field indicate male employees and which elements indicate female employees. The second line of the heading indicates which elements of the field correspond to the different values of *EEO* (*W*, *B*, *O*, *M*, *H*, and *A*). The third line of the heading consists of twelve groups of three minus signs (-) which are used simply to highlight the position in the report of each of the twelve elements of *SEXEEO*. When you have finished entering the column heading, you type *STOP* to continue with the next prompt.

```
01:GR
* GENERATE REPORT.
* CONSTRAINTS: ALL
   196 FOUND.
* PRINT FIELDS: (EXEMPT): ,SEXEEO:,ΔCOUNT:NUMBER:OF:EMPLOYEES
* ENTER HEADING FOR SEXEEO; ENTER STOP TO END:
:|---------MALE----------|--------FEMALE---------
:|  W   B   O   M   H   A|  W   B   O   M   H   A
: --- --- --- --- --- --- --- --- --- --- --- ---
:STOP
* SORT FIELDS: EXEMPT
* TOTAL FIELDS: SEXEEO,ΔCOUNT
* PRINT RECORD DETAILS? (Y/N): N
* REPORT WIDTH IS 70 CHARACTERS.
* ANY SPECIAL OPTIONS? (Y/N): N
* ALIGN PAPER, TYPE SPACE, CR TO START.
```

```
DATE:  9/08/83                                                    PAGE 1
                              01: PERSONNEL
                          CONSTRAINTS: ALL

            |---------MALE----------|--------FEMALE--------- NUMBER
            |  W   B   O   M   H   A|  W   B   O   M   H   A OF
             --- --- --- --- --- --- --- --- --- --- --- --- EMPLOYEES

NONEXEMPT    39  18   2  24  14   1  12   6   4  14   2   1       137
EXEMPT       12   6   0  14   6   0  10   4   1   3   3   0        59
            ------------------------------------------------- ---------
             51  24   2  38  20   1  22  10   5  17   5   1       196

* CONSTRAINTS: STOP
01:STOP
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

Because *EXEMPT* was specified as the *SORT FIELD* and no record details were requested, the report consists of only two lines—one for each value of *EXEMPT*. As requested, the code descriptions for the *EXEMPT* field appear as the labels for each row of figures.

By subtotalling the twelve-element binary field, *SEXEEO*, you effectively cross-tabulated the fields *SEX* and *EEO* with *EXEMPT*. By totalling *ΔCOUNT* (which has the column heading *NUMBER OF EMPLOYEES*), the total number of employees with each value of *EXEMPT* is also included in the report.

The following table briefly describes the different types of report that are produced depending on the responses given for *SORT FIELDS*, *TOTAL FIELDS*, and *PRINT RECORD DETAILS*.

Description of Report	*SORT FIELDS*	*TOTAL FIELDS*	*PRINT RECORD DETAILS?*
A simple, unsorted listing of records	*NONE*	*NONE*	*Y*
Grand totals only	*NONE*	fields	*N*
Unsorted listing with grand totals	*NONE*	fields	*Y*
First record in each break-field's group	fields	*NONE*	*N*
A sorted listing	fields	*NONE*	*Y*
Subtotals and grand totals without record details	fields	fields	*N*
Subtotals and grand totals with record details	fields	fields	*Y*

The organization and format of your report is further affected by the use of two special symbols in the specification of *SORT FIELDS*. The **sortbreak** symbol (/) and the **pagebreak** symbol, when used in place of the regular delimiter between field names (,), control grouping, totalling, and paging of the report. Use of a sortbreak slash (/) means that records are to be grouped and, if *TOTAL FIELDS* are specified, subtotalled for each different combination of values that occurs for the fields

to the left of the slash. Fields (if any) to the right of the slash control sorting within each sortbreak group, but do not result in grouping or subtotalling.

Use of a pagebreak star (*) results in the report starting on a new page for each different combination of values that occurs for fields to the left of the star. Because all records listed on a particular page have the same values for fields to the left of the star, these field values are removed from the columns of the report. Instead, the common field values are included in the page heading. The pagebreak star causes totalling to occur automatically at each pagebreak. You can also have sortbreaks occur within each page of the report by including the sortbreak slash between fields to the right of the star.

The following is a summary of the different ways *SORT FIELDS* can be specified and the resulting effect on the report.

fld1/ Groups records (and subtotals if *TOTAL FIELDS* are specified) according to each different value of **fld1**.

fld1,fld2/fld3 Groups records (and subtotals) for each different combination of values for **fld1** and **fld2**; within each category, sorts (but does not group or total) by **fld3**.

fld1*fld2/fld3 Produces a pagebreak for each different value of **fld1**; groups records (and totals) for each different combination of values for **fld1** and **fld2**; within each group, sorts (but does not group or subtotal) by **fld3**.

fld1*fld2 Produces pagebreaks for each different value of **fld1**; within each page, sorts (but does not group or subtotal) by **fld2**.

The following is a summary of the special options available with *GR*.

When *GR* prompts for *SPECIAL OPTIONS*, you can respond with any one or any combination of the following:

BATCHPRINT If you select the *BATCHPRINT* option, MABRA uses a batch task to generate your report. This means your terminal can be used for other work while the report is being generated. So, particularly for large reports, this option allows you to make more efficient use of the computer and communications resources. If sorting is required, the FILESORT utility is used, which means you can include more records in the report—up to the maximum number of records in your MABRA system. Once generated, the report is printed on the specified I.P. Sharp highspeed printer, using the SHARP APL Highspeed Print utility.

If you choose this option, you are prompted for print specifications and delivery instructions. Then you must indicate whether your print request should be processed by an N-task or a B-task. Once you have confirmed that the batch print should be submitted, the print file number and the N-task or B-task number are displayed.

DEPTH You use the *DEPTH* option to change the page depth and the print depth of your report. The page depth, usually referred to as form depth, is the length of one page in print lines (the default is 66 lines). The print depth is the number of lines that print on one page (the default is 60 lines).

PAUSE When this option is selected, a pause will occur after each page of your report is printed. This gives you the opportunity to align the paper correctly before

the next page is printed. It also allows you to view your report more conveniently if it is being displayed on a CRT terminal. You press the SPACE bar and the RETURN key to resume the printing of a report.

SPACING This option lets you select *SINGLE*, *DOUBLE*, or *TRIPLE* spacing between lines of a report (the default is *SINGLE*).

TITLE When you select the *TITLE* option, you are prompted one line at a time to enter a report title. If you want a line of text to be centred on a line when it is printed, you include a shriek *C* (*!C*) at the end of the line of text. If you are using an APL keyboard, you type shriek by first typing ".", then pressing the backspace key and typing "'" (upper case *K*).

WIDTH The *WIDTH* option allows you to increase the width of your report. Increasing the width affects the centring of the page heading and the title.

Linking Data From Other MABRA Systems

To generate some reports, you will find it necessary to combine data from two or more MABRA systems. The *LS* (Link Systems) command lets you copy fields from other MABRA systems into the currently entered system. The following examples illustrate the different options associated with linking MABRA systems. System 10 (*EDUCATION*) and system 20 (*COURSES*), as well as system 1, are used in these examples.

First, however, to become familiar with systems 10 and 20, each system in turn is entered and their fields and records are listed. The *LR* (List Records) command was described earlier in this chapter, while *LF* (List Fields) is discussed in detail in Chapter 6. Briefly, when you enter the *LF* command, you are asked to enter the names of the fields you are interested in, as well as the attributes (name, type, etc.) of the fields that you wish to have displayed. In this case, all fields and all their attributes are printed.

```
* SYSTEM MODE
:ENTER;20
* LAST UPDATE: 28/07/83 19:35:24; CPU USED: 3
* NUMBER OF COMPLETED UPDATES: 2
* DATA MODE, SYSTEM IS 20: COURSES
20:LF
* LIST FIELD DEFINITIONS
* FIELDS: ALL
* ATTRIBUTES: ALL
* ALIGN PAPER, TYPE SPACE, CR TO START.
                                                    DATE:  9/08/83
                                20: COURSES

CLASS     NAME      TYPE   WIDTH INV  DESC  BYTES USED  FMT   EXPR
                                      WID   MAS   INV
-----     ----      ----   ----- ---  ----  ----- ----  ---   ----
REAL      NAME      CHAR      30        30.0        30A1
          NUMBER    INTE       1         4.0        1I10
          CREDITS   INTE       1         4.0        I2
SESSION   Δ         INTE       1                    I3    (NUM,1)ρRCDS/ιρRCDS
PSEUDO    ΔCOUNT    BINA       1                    I1    1
```

```
20:LR
* LIST RECORDS
* CONSTRAINTS: ALL
    10 FOUND.
* PRINT FIELDS: ALL
* SORT FIELDS: NUMBER
* ALIGN PAPER, TYPE SPACE, CR TO START.

NAME                                NUMBER CREDITS

BASIC COMPANY POLICIES                1001       3
BUILDING MAINTENANCE                  1101       5
MACHINE MAINTENANCE                   1105       5
SELLING                               1202       8
CUSTOMER RELATIONS                    1206       5
CONFLICT RESOLUTION                   1220      10
CORPORATE PLANNING                    1231      10
DATA PROCESSING SYSTEM                1440       8
ASSEMBLY                              1502       5
WAREHOUSING                           1505       5

* CONSTRAINTS: STOP
20:STOP
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

System 20 (*COURSES*) has three real (as opposed to pseudo) fields. *NAME* contains the different course names, *NUMBER* contains the course numbers, and the values stored in *CREDITS* indicate how many educational credits the courses are worth. From the list of records, you can see that the company offers 10 different courses, worth from 3 to 10 credits upon successful completion by an employee.

```
:ENTER;10
* LAST UPDATE: 29/07/83 10:13:14; CPU USED: 3
* NUMBER OF COMPLETED UPDATES: 3
* DATA MODE, SYSTEM IS 10: EDUCATION
10:LF
* LIST FIELD DEFINITIONS
* FIELDS: ALL
* ATTRIBUTES: ALL
* ALIGN PAPER, TYPE SPACE, CR TO START.
                                                    DATE: 9/08/83
                          10: EDUCATION
CLASS     NAME      TYPE  WIDTH  INV  DESC  BYTES USED  FMT          EXPR
                                      WID   MAS   INV
-----     ----      ----  -----  ---  ----  ----- ----  ---          ----
REAL      EMPLOYEE  CHAR      4          4.0         4A1
          DATE      DATE      1          4.0         1G□ZZ/ZZ/Z9□
          COURSE    INTE      1          4.0         1I10

SESSION   Δ         INTE      1                      I3           (NUM,1)ρRCDS/⍳ρRCDS
PSEUDO    ΔCOUNT    BINA      1                      I1           1
```

```
10:LR
* LIST RECORDS
* CONSTRAINTS: ALL
   200 FOUND.
* PRINT FIELDS: ALL
* SORT FIELDS: COURSE
* ALIGN PAPER, TYPE SPACE, CR TO START.

EMPLOYEE      DATE      COURSE

0779           9/05/82        1001
0557          16/08/73        1001
0705           1/07/74        1001
9784          15/06/83        1001
5872           7/08/53        1001
5204          15/11/79        1001
8027           3/11/76        1001
9572          15/08/62        1001
2520          12/10/77        1001
9678           7/07/78        1001
9507          12/03/74        1001
5227          14/12/78        1001
5247           9/09/70        1001
7024           5/01/80        1001
4294          12/07/54        1001
6688           3/12/68        1001
7748           3/04/80        1001
4299          12/08/81        1001
6046           5/02/64        1001
5675           2/09/82        1001

.
.
.

* CONSTRAINTS: STOP
10:STOP
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

Each time an employee successfully completes a company course, a record is added to system 10. This system also has three real fields. *EMPLOYEE* contains the identification number of the employee who took the course. (In system 1, the same data is stored in a field called *IDNUMBER*.) *DATE* contains the date when the employee took the course, and *COURSE* contains the number of the course taken.

In the next part of this example, system 1 is entered and *LS* is used to bring in a field from system

10. System 1 contains administrative information about each employee, while system 10 contains information about the course completions of employees. With *LS*, a pseudo field is created to contain the total number of courses taken by each employee during a certain period.

When you bring in fields from another system with *LS*, MABRA matches records in the other system with records in the currently entered system. System 1 and system 10 both have fields that contain employee identification numbers. These fields, named *IDNUMBER* in system 1 and *EMPLOYEE* in system 10, serve to identify which course completion records in system 10 belong to which employee records in system 1. When MABRA brings in a field from system 10, the records are rearranged and matched up with the records in the current system by comparing the employee number fields in the two systems. When you are linking information from another MABRA system, the fields that match the records in the two systems are called **key fields**.

```
:ENTER;1
* LAST UPDATE:  7/08/83 14:06:23; CPU USED: 12
* NUMBER OF COMPLETED UPDATES: 4
* DATA MODE, SYSTEM IS 01: PERSONNEL
01:LS
* LINK SYSTEMS
* CREATOR AND SYSTEM NUMBER: 10
* TOTALLING REQUIRED? (Y/N): Y
* KEY FIELD IN CURRENT SYSTEM: IDNUMBER
* KEY FIELD IN SYSTEM TO BE LINKED: EMPLOYEE
* RECORD CONSTRAINT IN SYSTEM TO BE LINKED: DATE=82
    22 FOUND.
* FIELDS TO LINK FROM OTHER SYSTEM: ΔCOUNT
* NEW FIELD NAMES IN CURRENT SYSTEM: COURSES82
* OK TO LINK? (Y/N): Y
* 186 RECORDS IN CURRENT SYSTEM DID NOT MATCH RECORDS IN LINKED SYSTEM
  (ENTER PREVIOUS AT NEXT CONSTRAINTS PROMPT TO IDENTIFY UNMATCHED RECORDS)
* SYSTEMS LINKED.
01:
```

The first thing you are prompted for after you enter *LS* is the creator and system number of the MABRA system that contains the field(s) you wish to link. If the system has the same creator as the currently entered system, you only need to supply the system number.

You then must decide whether or not totalling is required. In this instance, field values need to be totalled as they are brought into system 1 from system 10. Totalling is required when the other system can have more than one record with the same key value (employee ID numbers, in this case). Remember that system 10 can have zero, one, or more course completion records for each employee. To match these records with the records in system 1 (which has only one record for each employee), you need to total the records by employee. In general, you need to request totalling with *LS* whenever the same key values can occur in more than one record in the other system; that is, when the key field values are not unique in the system being linked.

Next, you give the name of the key field in both the current system and the system to be linked. Although key fields do not have to have the same name in both systems, they must be the same data type and have the same field width.

LS then prompts you for a record constraint that defines which records from the other system are to be linked. You can have all records from the other system linked by entering *ALL* as the record constraint, or you can give a constraint that defines a subset of the records. In this example, only those records with a date value ending in 82 are linked.

Then you specify the name of the field(s) that you want to link from the other system. What takes place when *LS* actually links a field is that a session pseudo field is created in the current system to contain the values from the linked field. You must select a name for each pseudo field that *LS* creates (i.e., a name corresponding to each field linked from the other system). The name selected must not conflict with the name of any other field in the current system.

In this example, *ΔCOUNT* is the only field linked from the other system. This is the field that contains the value 1 for every record. By linking *ΔCOUNT* and requesting totalling, you cause *LS* to subtotal *ΔCOUNT* by employee for the records from system 10 that meet the constraint *DATE*=82.

The name *COURSES*82 is assigned to the session pseudo field that is created in the current system.

If you are satisfied with all your *LS* specifications, you would enter *Y* when MABRA asks if it can proceed with the linking. Depending on the size and complexity of the linking task that has been requested, the process may take anywhere from a few seconds to several minutes.

MABRA informs you that 186 records in the current system did not have matching records in the linked system, taking into account the constraints specified when linking. This means that 186 employees did not take any courses in 1982, which is not unusual. The newly-created *COURSES*82 field is assigned fill values (zeros, in this case) for the records of these employees. To list the records which did not match, you would use the *LR* command and specify *PREVIOUS* when prompted for *CONSTRAINTS*.

Listing the field definition for the session pseudo field *COURSES*82 gives the following information for that field.

```
01:LF
* LIST FIELD DEFINITIONS
* FIELDS: COURSES82
* ATTRIBUTES: ALL
* ALIGN PAPER, TYPE SPACE, CR TO START.

                                                    DATE:  9/08/83

                                  01: PERSONNEL

CLASS     NAME        TYPE  WIDTH INV  DESC  BYTES USED  FMT   EXPR
                                       WID   MAS   INV
-----     ----        ----  ----- ---  ----  ----- ----  ---   ----
SESSION   COURSES82   INTE      1                        I3    GDLINK 4381  7  1
PSEUDO
```

*COURSES*82 has the same width as *ΔCOUNT*, the field from system 10 which it was derived from. However, the data type and format statement have been adjusted to allow for totalling. The expression *GDLINK* 4381 7 1 is an APL statement that MABRA uses to retrieve linked data when you refer to this field.

Now the values from the linked field *COURSES*82 are listed together with the values from other fields already existing in the current system. Pseudo fields created using *LS* can be used in the same way as other fields—they can be used in constraints, or they can be specified as *PRINT FIELDS* or *SORT FIELDS* when reports are being defined.

```
01:LR
* LIST RECORDS
* CONSTRAINTS: COURSES82≠0
```

```
    10 FOUND.
* PRINT FIELDS: LASTNAME,FIRSTNAME,COURSES82
* SORT FIELDS: -COURSES82
* ALIGN PAPER, TYPE SPACE, CR TO START.

LASTNAME            FIRSTNAME    COURSES82

POWELL              DAVID                4
WALSH               WALTER               4
KING                MARTIN               4
PIKE                JOHN                 2
BLACKMAN            DAVID                2
BOOTH               ANDREW               2
ROBERTS             GEORGE               1
WARD                EDWARD               1
JOHNSON             DOROTHY              1
BLACKMAN            FRED                 1
```

As the record of any employee who did not take a course in 1982 would have a *COURSES82* value of zero, the constraint *COURSES82≠0* eliminates these records from the listing. The listing, which is sorted in descending order by the number of courses, shows the number of courses taken by each employee who took a course in 1982.

After the above listing is printed, you are prompted again for *CONSTRAINTS*. Since the next report to be produced involves linking system 10 (*EDUCATION*) with both system 20 (*COURSES*) and system 1 (*PERSONNEL*), you now exit system 1 and enter system 10.

```
* CONSTRAINTS: STOP
01:STOP
* MABRA VERSION 3.1
* SYSTEM MODE
:ENTER;10
* LAST UPDATE: 29/07/83 10:13:14; CPU USED: 3
* NUMBER OF COMPLETED UPDATES: 3
* DATA MODE, SYSTEM IS 10: EDUCATION
10:
```

If you return to system 1 again during the MABRA session, you will find that the pseudo field *COURSES82* is still present in that system. This is because session pseudo fields, including those created using *LS*, remain in your MABRA system for the duration of your MABRA session (even if you leave one system and enter another). Session pseudo fields are erased only when you exit from the MABRA program and return to APL immediate execution mode.

You can link one or more fields from the same MABRA system or from different systems, using the same or different key fields, as many times as necessary during a single MABRA session to combine the necessary data for reporting. In the next two examples, *LS* is used to link fields into system 10 from both system 20 and system 1. The records in system 10 contain data describing the course completions of employees, while the records in system 20 contain information on each company course. System 20 is being linked to obtain additional details about the courses an employee has taken (the course name and the number of credits a course is worth).

```
10:LS
* LINK SYSTEMS
* CREATOR AND SYSTEM NUMBER: 20
```

```
* TOTALLING REQUIRED? (Y/N): N
* KEY FIELD IN CURRENT SYSTEM: COURSE
* KEY FIELD IN SYSTEM TO BE LINKED: NUMBER
* RECORD CONSTRAINT IN SYSTEM TO BE LINKED: ALL
    10 FOUND.
* FIELDS TO LINK FROM OTHER SYSTEM: NAME,CREDITS
* NEW FIELD NAMES IN CURRENT SYSTEM: COURSENAME,CREDITS
* OK TO LINK? (Y/N): Y
* SYSTEMS LINKED.
10:
```

Because system 20 has the same creator as system 10, only the system number needs to be entered in response to the prompt for *CREATOR AND SYSTEM NUMBER*.

When each different value in the key field from the system to be linked occurs only once (i.e., is unique), you use *LS without* totalling. The field values from the other system are then *duplicated* for each matching record in the current system. In this example, the course information from system 20 is duplicated for each course enrolment record in the current system. Also, when you specify that totalling is not required, you can link character as well as numeric fields, as is done in this case.

The key fields in this example are the fields from each system that contain course numbers. In system 10 the field is called *COURSE*, while in system 20, the field is called *NUMBER*. Both fields have the same width and data type, and values in the two fields identify matching records.

By entering the constraint *ALL*, MABRA will link all records from system 20. The fields linked from system 20 are *NAME* and *CREDITS*. *NAME* is a character field that contains the full name of company courses, while *CREDITS* is a numeric field that contains the number of credits that each course is worth. The new pseudo fields that are created in system 10 to contain the linked field values are given the names *COURSENAME* and *CREDITS*.

Once the systems are linked, *COURSENAME* and *CREDITS* are created as session pseudo fields in system 10. These fields contain the values from *COURSE* and *CREDITS*, respectively, duplicated from system 20 for each matching record in the current system.

One of MABRA's most important features for time- and step-saving (not to mention cost-saving) is the stored answer-ahead facility. You will no doubt find, if you use MABRA on a regular basis, that there are certain commands you use repeatedly with the same specifications. To avoid typing the same specifications over again every time you use a command, you create a **stored answer-ahead**. This MABRA facility allows you to easily **store** and **recall** your answers to a series of MABRA prompts by assigning a name to the series of answers.

You begin saving your system entries by typing a left brace ({), the name you are giving to the series of answers, and a colon (:). MABRA responds that you are in the process of saving a series of entries under the name you specified, and you proceed to enter your responses to the MABRA prompts in the normal manner.

The following example creates a stored answer-ahead called *LINK1* which includes the *LS* command and a series of related responses.

```
10:{LINK1:
* ANSWER AHEAD DEFINITION OPENED FOR {LINK1}.
10:LS
* LINK SYSTEMS
```

```
* CREATOR AND SYSTEM NUMBER: 1
* TOTALLING REQUIRED? (Y/N): N
* KEY FIELD IN CURRENT SYSTEM: EMPLOYEE
* KEY FIELD IN SYSTEM TO BE LINKED: IDNUMBER
* RECORD CONSTRAINT IN SYSTEM TO BE LINKED: ALL
  196 FOUND.
* FIELDS TO LINK FROM OTHER SYSTEM: LASTNAME,FIRSTNAME,INITIAL
* NEW FIELD NAMES IN CURRENT SYSTEM: LASTNAME,FIRSTNAME,INITIAL
* OK TO LINK? (Y/N): Y
* 173 RECORDS IN LINKED SYSTEM DID NOT MATCH RECORDS IN CURRENT SYSTEM
* SYSTEMS LINKED.
10:}
* ANSWER AHEAD {LINK1} DEFINED.
10:
```

To terminate a stored answer-ahead string of entries, you type a right brace (}). MABRA then responds that your string of entries has been saved under the name you selected.

In this execution of *LS*, fields from system 1 (which contains employee data) are linked with system 10. Again, totalling is not requested because the employee information from system 1 is to be duplicated for each course enrolment record in system 10.

Records in the two systems are to be matched by comparing employee numbers, which are found in the *EMPLOYEE* field in the current system and the *IDNUMBER* field in the system to be linked.

The record constraint *ALL* instructs MABRA to link field values from all records in system 1. The fields that are to be linked from system 1 are *LASTNAME*, *FIRSTNAME*, and *INITIAL*. The same field names are assigned to the session pseudo fields that must be created in system 10 to contain the data that is duplicated from these linked fields.

Once the two systems have been linked, MABRA reports that 173 records from the other system did not match any records in the current system. This indicates that 173 employees have never completed any courses. At this point, *LASTNAME*, *FIRSTNAME*, and *INITIAL* exist as session pseudo fields in system 10. The link as well as these fields will remain until you exit from the MABRA program.

To display the series of entries stored in an answer-ahead definition, you type a left brace, a question mark, the name of the answer-ahead definition, and a right brace, as follows:

```
10:{?LINK1}
{LINK1   } LS;1;N;EMPLOYEE;IDNUMBER;ALL;LASTNAME,FIRSTNAME,INITIAL
          ;LASTNAME,FIRSTNAME,INITIAL;Y
```

Although the session pseudo fields created by this series of entries are erased when you exit from the MABRA program, they can be created again very simply in future MABRA sessions by recalling the answer-ahead definition *LINK1*. To recall a stored answer-ahead, you type a left brace, the name of the answer-ahead, and a right brace, as follows:

```
10:{LINK1}
```

MABRA will retrieve the stored series of answers and enter them as if you typed them yourself. The *LS* command will therefore be executed with the same set of answers you typed while defining *LINK1*.

You can start and end a stored answer-ahead definition at any prompt within *DATA MODE*. Then, it

must be recalled at the same prompt where you began the definition. Answer-ahead definitions are stored and used with one particular MABRA system. If they are defined by the account that is the creator account for the system, the stored answer-aheads are retained for future MABRA sessions and are available to all users who have access to the system. If a user does not have access to a command or field referred to in a stored answer-ahead, or if the answer-ahead is invalid for any other reason (e.g., it refers to a field that has been deleted), appropriate error messages will be displayed when the answer-ahead is executed.

To change a stored answer-ahead, you redefine it using the same name. When you save an answer-ahead with a name that is already in use, the new definition replaces the old one.

The following table summarizes the use of stored answer-ahead definitions:

{name:entry; . . .}	Save a series of MABRA entries as a stored answer-ahead definition called **name**.
{name}	Recall and enter the series of MABRA entries that is stored as the answer-ahead **name**.
{?name}	Display the series of entries that is saved as the stored answer-ahead **name**.
{?}	Display all the answer-aheads that are stored with the currently entered MABRA system together with their definitions.
{?string}	Display the stored answer-ahead definitions whose names begin with **string**.
{*name}	Delete the stored answer-ahead **name**.

In the next example, another session pseudo field is added to system 10—this time using the *AP* command. The field is to be derived from the other session pseudo fields created by *LS—LASTNAME*, *FIRSTNAME*, and *INITIAL*. Here again, responses are saved in a stored answer-ahead definition.

```
10:AP
* ADD PSEUDO FIELD
* ENTER FIELD NAME: {APNAME:NAME;CHAR;40;Y
* ANSWER AHEAD DEFINITION OPENED FOR {APNAME}.
* ENTER EXPRESSION DEFINING FIELD
:RXB(NUM,40)↑FIRSTNAME,' ',INITIAL,' ',LASTNAME
* RETAIN THIS FIELD? (Y/N): N
* OK TO ADD FIELD? (Y/N): Y}
* ANSWER AHEAD {APNAME} DEFINED.
* PSEUDO FIELD ADDED.
* ENTER FIELD NAME: STOP
10:
```

When you are prompted to enter a field name, you first type {*APNAME*: to begin defining a stored answer-ahead called *APNAME*. Then, on the same line, you enter four responses to *AP* prompts. This illustrates the MABRA feature of "pre-answering" which you can do whenever you know exactly what prompts are associated with a particular command. Responses must be separated by a semi-colon (;).

The field that is being created is to be called *NAME*. It is to contain 40 elements of character data,

and it is to be output in the default format. The field values to be stored in *NAME* are a combination of the *FIRSTNAME*, *INITIAL*, and *LASTNAME* values with extra (redundant) blanks removed. An APL expression that performs this data manipulation is entered. The field cannot be retained, as the fields from which it is derived are session pseudo fields.

When you are sure that your specifications have been entered correctly, you confirm that MABRA can proceed to add the pseudo field. In this case, you also terminate the stored answer-ahead at this point by immediately following the *Y* by a right brace. MABRA responds that *APNAME* has been defined and the pseudo field has been added. When *AP* prompts for another pseudo field name, you enter *STOP* to exit from the command.

After linking two fields from system 20 and three fields from system 1, and after adding a pseudo field to combine the three fields from system 1 into one field, a listing of all the fields in system 10 may be useful.

```
10:LF
* LIST FIELD DEFINITIONS
* FIELDS: ALL
* ATTRIBUTES: ALL
* ALIGN PAPER, TYPE SPACE, CR TO START.

                                                    DATE:  9/08/83

                                    10: EDUCATION

CLASS     NAME         TYPE  WIDTH  INV  DESC  BYTES  USED  FMT          EXPR
                                          WID   MAS    INV
-----     ----         ----  -----  ---  ----  -----  ----  ---          ----
REAL      EMPLOYEE     CHAR      4            4.0          4A1
          DATE         DATE      1            4.0          1G□ZZ/ZZ/Z9□
          COURSE       INTE      1            4.0          1I10

SESSION   Δ            INTE      1                         I3           (NUM,1)⍴RCDS/⍳⍴RCDS
PSEUDO    ΔCOUNT       BINA      1                         I1           1
          COURSENAME   CHAR     30                         30A1         GDLINK  908   10   30
          CREDITS      INTE      1                         I2           GDLINK 6762   11    1
          LASTNAME     CHAR     20                         20A1         GDLINK  1357    14    20
          FIRSTNAME    CHAR     15                         15A1         GDLINK  1797    15    15
          INITIAL      CHAR      1                         1A1          GDLINK 26936    16     1
          NAME         CHAR     40                         40A1         RXE(NUM,40)↑ΔFIRSTNAME,' ',ΔINITIAL,' ',ΔLASTNAME
```

This listing describes the three real fields that always existed in system 10, followed by eight session pseudo fields. The fields Δ and Δ*COUNT* always occur when you enter any system. *COURSENAME* and *CREDITS* were added by linking with system 20, while *LASTNAME*, *FIRSTNAME*, and *INITIAL* were added when system 1 was linked. The pseudo field *NAME*, a combination of the fields *LASTNAME*, *FIRSTNAME*, and *INITIAL*, was created using *AP*.

Notice in the expression listed for *NAME* that MABRA adds Δ (*delta-underscore*) to the field names *FIRSTNAME*, *INITIAL*, and *LASTNAME*. MABRA puts Δ in front of field names that are used in a pseudo field's APL expression. MABRA makes this and other adjustments in order to process the expression. The MABRA *Programmer's Guide* explains more about Δ.

Now that all these pseudo fields have been created, you are ready to generate a report. This report will show the courses each employee completed in 1982 and the number of credits earned.

```
10:GR
* GENERATE REPORT.
* CONSTRAINTS: DATE=82
    22 FOUND.
* PRINT FIELDS: NAME:EMPLOYEE,COURSE,COURSENAME,CREDITS
* SORT FIELDS: NAME*COURSE
* TOTAL FIELDS: CREDITS
* PRINT RECORD DETAILS? (Y/N): Y
* REPORT WIDTH IS 50 CHARACTERS.
* ANY SPECIAL OPTIONS? (Y/N): Y

* OPTION: TITLE
* ENTER TITLE (ENTER STOP TO END):
COMPANY COURSES TAKEN BY EMPLOYEE IN 1982!C
WITH TOTAL CREDITS!C
STOP

* OPTION: DEPTH
* PAGE DEPTH: 22
* PRINT DEPTH: 20

* OPTION: PAUSE
* PAUSE AFTER EACH PAGE? (Y/N): Y

* OPTION: STOP
* ALIGN PAPER, TYPE SPACE, CR TO START.
```

The constraint *DATE*=82 is entered because the report is only supposed to show courses completed by employees in 1982. When prompted for *PRINT FIELDS*, you specify all the fields that you wish to include in the report (in the order that you wish to have them appear), and you also specify an alternate column heading for the *NAME* field.

The records included in this report are to be sorted alphabetically by *NAME*, and, within *NAME*, they are to be sorted in ascending order according to the value of *COURSE*. Typing * to the right of *NAME* instructs *GR* to start a new page each time the value of *NAME* changes, so there will be a separate page for each employee.

By specifying *CREDITS* as a *TOTAL FIELD*, the report will show the total number of credits earned by each employee. Record details are requested for this report.

Finally, a number of special options are required for more control over the printing of the report. First, you want to enter a title for the report. *GR* instructs you to enter your title, one line at a time, and to type *STOP* when you are finished. You enter a two-line title and indicate that each line is to be centred by typing *!C* at the end of each line.

Next, you want to change the report depth. By assigning a form depth of 22, you tell *GR* that a page

now consists of 22 lines (instead of the usual 66). A print depth of 20 tells *GR* that a maximum of 20 lines can be printed on one page. The difference between form depth and print depth is the number of blank lines left at the bottom of each page.

The other special option that is required for this report is *PAUSE*, which causes *GR* to stop printing after each page. This allows the form or paper to be aligned properly before the next page of the report is printed.

If you have specified all the special options needed for the report, you enter *STOP* when *GR* prompts for another *OPTION*.

The report produced by these specifications is actually 11 pages in length, but only the first three pages and the last page are shown here. Each of the first 10 pages lists the courses taken by a different employee in 1982 and the number of credits earned, while the last page shows the total value of *CREDITS* (the total for all employees). The current date and a page number appear at the top of each page followed by the report title.

```
DATE:  9/08/83                                    PAGE 1
     COMPANY COURSES TAKEN BY EMPLOYEE IN 1982
                WITH TOTAL CREDITS

EMPLOYEE=ANDREW P BOOTH

    COURSE COURSENAME                          CREDITS

      1231 CORPORATE PLANNING                       10
      1440 DATA PROCESSING SYSTEM                    8
                                                  -------
                                                     18
```

```
DATE:  9/08/83                                    PAGE 2
     COMPANY COURSES TAKEN BY EMPLOYEE IN 1982
                WITH TOTAL CREDITS

EMPLOYEE=DAVID A POWELL

    COURSE COURSENAME                          CREDITS

      1206 CUSTOMER RELATIONS                        5
      1220 CONFLICT RESOLUTION                      10
      1231 CORPORATE PLANNING                       10
      1440 DATA PROCESSING SYSTEM                    8
                                                  -------
                                                     33
```

```
DATE:  9/08/83                                    PAGE 3
     COMPANY COURSES TAKEN BY EMPLOYEE IN 1982
               WITH TOTAL CREDITS

EMPLOYEE=DAVID R BLACKMAN

    COURSE COURSENAME                          CREDITS

      1220 CONFLICT RESOLUTION                      10
      1231 CORPORATE PLANNING                       10
                                                   -------
                                                    20
```

```
.
.
.
```

```
DATE:  9/08/83                                    PAGE 11
     COMPANY COURSES TAKEN BY EMPLOYEE IN 1982
               WITH TOTAL CREDITS

[TOTAL]

    COURSE COURSENAME                          CREDITS

                                                  168
```

```
* CONSTRAINTS: STOP
10:STOP
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

Specifying and Printing Standard Forms

The *SF* (Specify Form) command enables you to set up a standard form, and the *PF* (Print Form) command enables you to print a form that you have specified with *SF*. Other MABRA report commands allow you to print only one line per record. *SF* and *PF*, on the other hand, are used when you want to print certain fields from each record in a report having a matrix or template format. You use *SF* and *PF*, for example, to print mailing labels or to insert record information on pre-printed forms. You can also specify text combined with record values for printing things like form letters.

In the next example, system 1 is entered and fields are linked from system 10. Then *SF* and *PF* are used to produce address labels for letters that will be sent to employees. Finally, *SF* and *PF* are again used, this time to produce a form letter that will be addressed to all employees who have completed company courses. The letter will tell them how many course credits they earned in a particular year.

```
:ENTER
* SYSTEM NUMBER: 1
* BP BTASK 49270 SUBMITTED SORTREQ 21633
* BP BTASK 49270 SUBMITTED BTASK 49271
* BP BTASK 49270 COMPLETED AT 17:20:07 09/08/83
* BP BTASK 49271 SUBMITTED HREQ 362063
* LAST UPDATE:  7/08/83 14:06:23; CPU USED: 12
* NUMBER OF COMPLETED UPDATES: 4
* DATA MODE, SYSTEM IS 01: PERSONNEL
01:
```

This time when system 1 is entered, certain information is displayed about the record listing that was requested previously via *BP*. MABRA's B-task 49270 first instructed the SHARP APL system's SORTREQ program to sort the records to be included in the listing according to the *SORT FIELDS* that had been specified. This particular job was SORTREQ 21633. Then B-task 49270 submitted B-task 49271 to actually process the report, and 49270 completed. B-task 49271 proceeded to submit a request to the HSPRINT program to actually print the listing. This was submitted as request number 362063.

```
01:LS
* LINK SYSTEMS
* CREATOR AND SYSTEM NUMBER: 10
* TOTALLING REQUIRED? (Y/N): Y
* KEY FIELD IN CURRENT SYSTEM: IDNUMBER
* KEY FIELD IN SYSTEM TO BE LINKED: EMPLOYEE
* RECORD CONSTRAINT IN SYSTEM TO BE LINKED: DATE=82
    22 FOUND.
* FIELDS TO LINK FROM OTHER SYSTEM: CREDITS
* NEW FIELD NAMES IN CURRENT SYSTEM: CREDITS
* OK TO LINK? (Y/N): Y
* 186 RECORDS IN CURRENT SYSTEM DID NOT MATCH RECORDS IN LINKED SYSTEM
  (ENTER PREVIOUS AT NEXT CONSTRAINTS PROMPT TO IDENTIFY UNMATCHED RECORDS)
* SYSTEMS LINKED.
01:
```

The *LS* command is used with the specifications shown above to link system 1 with system 10, which contains the course enrolment records. Because you have not exited from the MABRA program since the fields *NAME* and *CREDITS* were linked from system 20 into system 10, these fields are still accessible as you now link system 1 with system 10.

The key field in system 1 is *IDNUMBER*, and in system 10 it is *EMPLOYEE*. Since many records in the

system to be linked have key field values that will match one record's key field value in the current system, the field values that are brought in from the system being linked must be totalled. This means that only numeric fields can be linked.

The constraint *DATE*=82 specifies that only those records from system 10 with a *DATE* value equal to 1982 are to be linked. System 10's session pseudo field *CREDITS* (which was previously linked from system 20 into system 10) is the field to be linked. This example illustrates that you can link fields in a chain through as many systems as necessary, as long as you do not exit from the MABRA program at any point in the linking process. The field name *CREDITS*, being appropriately descriptive, is assigned to the newly-created pseudo field.

Now the form for the mailing labels is set up with the *SF* command.

```
01:SF
* SPECIFY FORM
* THERE ARE 0 FORM SPECIFICATIONS
* LIST, DELETE, ADD, OR CHANGE? (L/D/A/C): A
* SPECIFICATION NUMBER WILL BE 1
* STANDARD 3-UP-LABEL FORM? (Y/N): Y
* FORM DEPTH IS 9.
* FORM WIDTH IS 41.
* 3 FORMS ACROSS PAGE.
* 8 FORMS DOWN PAGE.
* PAGE DEPTH IS 72.
```

When you enter the *SF* command, you are first told how many standard forms are currently specified for the system you are using. Then you are asked if you wish to list, delete, add, or change a form specification. In this example, you give the response required to add a new form specification. *SF* assigns an identification number to every form that is added. This is required so a form can be referred to when you want to list, change, delete, or print it.

Next, you are asked if the reports to be generated using this standard form are to be printed on a particular paper form that consists of three adhesive labels across a sheet ("3-up") and eight labels down the sheet. This is the format for address labels printed by the Sharp Highspeed Printing Service when you use the form specification *LB3U*. If you answer *Y*, *SF* automatically sets the dimensions that are required for this particular form type and then displays them. If you answer *N*, you are prompted for form dimensions, as you will see in the next example.

3-up labels have a form depth of 9 and a form width of 41. Form depth is the number of rows that each form consists of and form width is the number of columns. These labels are printed three across the page and eight down the page, so each page contains 24 labels. A page of label stock has 72 print lines.

The next step in setting up a form is defining the actual contents of the form. This involves specifying any literal text that is to appear and indicating where it is to be placed. Similarly, any field values that are to be included must be specified together with their position on the form. The different positions on the form are identified by row and column numbers. In setting up a form for mailing labels, normally only field values are included.

```
* FIELD OR TEXT? (F/T): F
* FIELD: FIRSTNAME
* ROW: 1
* COLUMN: 1
```

```
* FIELD OR TEXT? (F/T): F
* FIELD: LASTNAME
* ROW: 1
* COLUMN: 17

* FIELD OR TEXT? (F/T): F
* FIELD: ADDRESS1
* ROW: 2
* COLUMN: 1

* FIELD OR TEXT? (F/T): F
* FIELD: ADDRESS2
* ROW: 3
* COLUMN: 1

* FIELD OR TEXT? (F/T): F
* FIELD: CITY
* ROW: 4
* COLUMN: 1

* FIELD OR TEXT? (F/T): F
* FIELD: ZIP
* ROW: 4
* COLUMN: 32

* FIELD OR TEXT? (F/T): STOP
* REMOVE BLANK ROWS? (Y/N): Y
* TRIM EXTRA BLANKS BETWEEN COLUMNS? (Y/N): Y
* WHICH ROWS CONTAIN THE COLUMNS?: 1 4
* OK TO UPDATE? (Y/N): Y
* UPDATED.
* LIST, DELETE, ADD, OR CHANGE? (L/D/A/C):
```

SF repeats the process of prompting for either field or textual input and its position until you enter *STOP* to indicate that you have finished defining the contents of the form.

This form, as you can see, is made up entirely of field values. For each field value to be included, the name of the field is entered plus its position as defined by row and column numbers.

FIRSTNAME field values are to be printed starting at column 1 of row 1. Since *FIRSTNAME* has a field width of 15, the starting position given for *LASTNAME* is column 17 of row 1 (allowing for one blank column between the two fields). Positions are also defined for the fields *ADDRESS*1, *ADDRESS*2, *CITY*, and *ZIP*.

After you have specified the contents of the form, *SF* asks if blank rows should be removed. This refers only to rows that are supposed to contain field values according to your form specification. For example, if a label was being produced using values from a record that contained a blank (null) *ADDRESS*1, the second row of that label would be entirely blank. However, if you had requested that blank rows be removed, the third and fourth rows would be moved up one line giving the label a neater appearance.

Removing extra (redundant) blanks from between field values also gives a neater appearance to forms. Fields that are inserted on a form are filled out to the full field width with blank spaces. For example, if a record contained the *FIRSTNAME* value *TOM*, the field would actually contain the three letters of

the name followed by 12 blanks. This is what would appear on the label if extra blanks were not removed. In this example, if extra blanks were trimmed from the row that contains *FIRSTNAME* followed by *LASTNAME*, all but one of the blanks between the first and last names would be removed. If you respond that extra blanks between columns should be trimmed, you are then asked to specify which rows they are to be trimmed from.

Finally, if you are satisfied that your specifications are correct, you tell *SF* to proceed with creating the form.

After processing the request, *SF* prompts for another form option. Since form specification 1 has now been created, it can be listed. A specification listing consists of two main parts—information about the form dimensions and the trimming of blank rows and spaces, and information about the form's contents.

```
* LIST, DELETE, ADD, OR CHANGE? (L/D/A/C): L
* LIST WHICH SPECIFICATION NUMBERS: 1
* ALIGN PAPER, TYPE SPACE, CR TO START.

SPECIFICATION NUMBER 1. FORM DIMENSIONS ARE 9 ROWS, 41 COLUMNS.
                        3 FORMS ACROSS PAGE.
                        8 FORMS DOWN PAGE.
                        PAGE DEPTH IS 72.
                        BLANK ROWS REMOVED.
                        TRIMMED ROWS ARE 1 4.

   ITEM  ---STARTS---    FIELD NAME
    NO.   ROW  COLUMN    OR 'TEXT'

     1     1     1       FIRSTNAME
     2     1    17       LASTNAME
     3     2     1       ADDRESS1
     4     3     1       ADDRESS2
     5     4     1       CITY
     6     4    32       ZIP

* LIST, DELETE, ADD, OR CHANGE? (L/D/A/C): STOP
01:
```

Each field to be included in the form is listed together with an item number which would be used when changing or editing the form specification. The items are listed in order by row and column numbers.

When you want to terminate the *SF* command, you type *STOP* in response to the *LIST, DELETE, ADD, OR CHANGE* prompt.

Now you use the *PF* command to print mailing labels using the form just set up with *SF*.

```
01:PF
* PRINT FORM
* PRINT WHICH FORM SPECIFICATION: 1
* CONSTRAINTS: 122 98 81 68 125 138
     6 FOUND.
* SORT FIELDS: LASTNAME,FIRSTNAME
* TERMINAL OR HIGH SPEED PRINT? (T/H): T
* PAUSE AFTER EACH PAGE? (Y/N): N
```

```
* ALIGN PAPER, TYPE SPACE, CR TO START.

CHARLES ALBERT              GARY ALEXANDER              ROBERT ALEXANDER
55 LINCOLN MILLS RD.        17 EMERALD POINT RD.        92 BARKER ROAD
PITTSBURGH, PA 41783        NORFOLK, VA 38748           BURLINGTON, VT 21093


SAMUEL ALEXANDER            MATHEW GILMORE              MARY GOODMAN
455 TITUS AVE.              15 ELM ST.                  601 LINDEN AVE.
GREENWICH, CT 11388         ROCHESTER, NY 14622         SUITE 444
                                                        NORFOLK, VA 38748
```

When you enter the *PF* command, you are first asked to enter the number of the form specification you want to print. In this example, when you are prompted for *CONSTRAINTS*, you select a limited number of records by entering Δ record numbers.

The labels can be sorted by any field or combination of fields. These labels will be printed alphabetically according to *LASTNAME* and *FIRSTNAME*.

In this case, you request that the printing be done at your terminal. You can have the printing pause after each page in order to align the paper properly before the next page is printed. You would want the printing to pause, for example, if you were inserting paper or forms in the terminal, one page at a time. If you are using continuous forms at a terminal with a form feed tractor, however, you would answer *N* to the prompt regarding pausing.

Another listing is now requested with *PF*.

```
* PRINT WHICH FORM SPECIFICATION: 1
* CONSTRAINTS: ALL
   196 FOUND.
* SORT FIELDS: LASTNAME,FIRSTNAME
* TERMINAL OR HIGH SPEED PRINT? (T/H): H
* NOTE: ENTER PRINT SPEC 'LB3U' FOR STANDARD 3-UP LABELS
* PRINT SPECS: LB3U
* ENTER DELIVERY INSTRUCTIONS. TYPE 'STOP' TO END INSTRUCTIONS.
DELIVER TO DISTRIBUTING COMPANY, INC.
IN ACCORDANCE WITH USUAL DELIVERY INSTRUCTIONS
ON FILE
STOP
* RUN AS N-TASK OR B-TASK? (N/B): N
* SUBMIT REQUEST? (Y/N): Y
* PRINT FILE IS: P102320052
* NTASK NO. 4986
* PRINT WHICH FORM SPECIFICATION: STOP
01:
```

This time you ask *PF* to print the report or listing on the highspeed printer. You are reminded that if the listing is to be printed on the special paper that contains three adhesive labels across and eight labels down each page, you must enter the print specification *LB3U*.

Delivery instructions are given, and *STOP* is entered to indicate the end of delivery specifications.

As with other report requests, MABRA can process it as an N-task or a B-task—N-tasks being the faster method and B-tasks being the more economical. An N-task is requested in this example.

Highspeed print requests are submitted by means of a temporary print file. MABRA gives you the name of the file it creates. In MABRA, this file is erased automatically when printing is complete unless you enter the print specification *NOERASE*.

Another form is created by the following MABRA specifications.

```
01:SF
* SPECIFY FORM
* THERE IS 1 FORM SPECIFICATION
* LIST, DELETE, ADD, OR CHANGE? (L/D/A/C): A
* SPECIFICATION NUMBER WILL BE 2
* STANDARD 3-UP-LABEL FORM? (Y/N): N
* FORM'S DEPTH: 66
* FORM'S WIDTH: 70
* NUMBER OF FORMS ACROSS THE PAGE: 1
* NUMBER OF FORMS DOWN THE PAGE: 1
* PAGE DEPTH (HIGH SPEED PRINT CARRIAGE SETTING): 66
```

This form is assigned the identification number 2. It is not going to be another 3-up-label form, but rather a full-page letter. You specify that the form depth is to be 66 lines. This is the number of lines that fit on a standard North American sheet of paper measuring 8.5 by 11 inches (21.6 by 28 cm.). You also want the form's printed width to be 70 columns. The standard page has 85 columns, so room for margins is allowed for. With these form dimensions, it will only be possible to print one letter across the page and one letter down the page, so this is what you must specify.

When you are prompted to enter *PAGE DEPTH*, you enter the number of lines that should fit on one page.

Having set the form's dimensions, you can now proceed to itemize *what* is to be printed in this form and *where*.

```
* FIELD OR TEXT? (F/T): T
* TEXT: JULY 31, 1983
* ROW: 12
* COLUMN: 45

* FIELD OR TEXT? (F/T): F
* FIELD: FIRSTNAME
* ROW: 14
* COLUMN: 1

* FIELD OR TEXT? (F/T): F
* FIELD: LASTNAME
* ROW: 14
* COLUMN: 17
* FIELD OR TEXT? (F/T): F
* FIELD: ADDRESS1
* ROW: 15
* COLUMN: 1
```

```
* FIELD OR TEXT? (F/T): F
* FIELD: ADDRESS2
* ROW: 16
* COLUMN: 1

* FIELD OR TEXT? (F/T): F
* FIELD: CITY
* ROW: 17
* COLUMN: 1

* FIELD OR TEXT? (F/T): F
* FIELD: ZIP
* ROW: 17
* COLUMN: 32

* FIELD OR TEXT? (F/T): T
* TEXT: DEAR
* ROW: 19
* COLUMN: 1

* FIELD OR TEXT? (F/T): F
* FIELD: FIRSTNAME
* ROW: 19
* COLUMN: 6

* FIELD OR TEXT? (F/T): T
* TEXT: :
* ROW: 19
* COLUMN: 21

* FIELD OR TEXT? (F/T): T
* TEXT: EVERY YEAR YOUR COMPANY PROVIDES YOU WITH A STATEMENT OF EMPLOYEE
* ROW: 21
* COLUMN: 6

* FIELD OR TEXT? (F/T): T
* TEXT: BENEFITS TO WHICH YOU ARE ENTITLED AS AN EMPLOYEE OF DISTRIBUTING
* ROW: 22
* COLUMN: 1

* FIELD OR TEXT? (F/T): T
* TEXT: COMPANY, INC. YOUR CURRENT SALARY IS $
* ROW: 23
* COLUMN: 1
* FIELD OR TEXT? (F/T): F
* FIELD: SALARY
* ROW: 23
* COLUMN: 39

* FIELD OR TEXT? (F/T): T
* TEXT: . IN ADDITION, YOU ARE
* ROW: 23
* COLUMN: 49
```

```
* FIELD OR TEXT? (F/T): T
* TEXT: ENTITLED MANY OTHER EMPLOYEE BENEFITS. IN THE PAST YEAR, YOU HAVE
* ROW: 45
* COLUMN: 1

* FIELD OR TEXT? (F/T): T
* TEXT: EARNED
* ROW: 25
* COLUMN: 1

* FIELD OR TEXT? (F/T): F
* FIELD: CREDITS
* ROW: 25
* COLUMN: 8

* FIELD OR TEXT? (F/T): T
* TEXT: CREDITS FOR TAKING COURSES IN THE COMPANY'S EDUCATIONAL
* ROW: 25
* COLUMN: 11

* FIELD OR TEXT? (F/T): T
* TEXT: PROGRAM.
* ROW: 26
* COLUMN: 1

* FIELD OR TEXT? (F/T): T
* TEXT: SINCERELY,
* ROW: 28
* COLUMN: 45

* FIELD OR TEXT? (F/T): T
* TEXT: DISTRIBUTING COMPANY, INC.
* ROW: 30
* COLUMN: 45

* FIELD OR TEXT? (F/T): STOP
* REMOVE BLANK ROWS? (Y/N): Y
* TRIM EXTRA BLANKS BETWEEN COLUMNS? (Y/N): Y
* WHICH ROWS CONTAIN THE COLUMNS?: 14 23 25
* OK TO UPDATE? (Y/N): Y
* UPDATED.
* LIST, DELETE, ADD, OR CHANGE? (L/D/A/C):
```

The first item specified for this form is a text item. Text items included in a form appear exactly as they are specified for every record that the form is printed for. In this case, every form that is printed will have the date *JULY 31, 1983* appearing in the upper right corner (on row 12 starting in column 45). The next six items are field items that make up the letter's inside address. This part of the letter resembles the mailing label format defined in form specification 1.

The next three items make up the salutation that is to appear on row 19 of the letter.

The ten items that follow are a combination of field and text items that define the body of the letter. The first line of the body is to be printed on line 21 (leaving one blank line between the salutation and the body) starting at column 6 (indented 5 spaces from the left margin):

When you are combining field and text items on the same line, you must make sure that the starting columns are all determined properly. For example, on row 23 field values for *SALARY* are to be inserted between two text strings. The text preceding *SALARY* is 38 characters in length, so the starting position for *SALARY* is column 39. The format statement used to print *SALARY* values indicates that it may take up to 10 columns to print a salary value. This means that the text following *SALARY* must begin at column 49.

The last two text items define the complimentary closing and signature for the letter.

The final step in setting up a form involves specifying whether or not blank rows are to be removed and extra blanks between columns are to be trimmed. For this form specification, blank rows are to be removed. However, only blank rows that should have contained field values according to the form specification are removed. Blank rows that are intentionally left blank according to the way the form has been set up (e.g., the blank line between the letter's inside address and the salutation) are **not** removed.

Also for this particular form, if a field's value does not fill its specified width, extra blank columns are to be omitted. Rows 14, 23, and 25 are the rows that may require this trimming.

Now that the second form specification has been defined, it can be listed.

```
* LIST, DELETE, ADD, OR CHANGE? (L/D/A/C): L
* LIST WHICH SPECIFICATION NUMBERS: 2
* ALIGN PAPER, TYPE SPACE, CR TO START.

SPECIFICATION NUMBER 2. FORM DIMENSIONS ARE 66 ROWS, 70 COLUMNS.
                        1 FORM ACROSS PAGE.
                        1 FORM DOWN PAGE.
                        PAGE DEPTH IS 66.
                        BLANK ROWS REMOVED.
                        TRIMMED ROWS ARE 14 23 25.

  ITEM  ---STARTS---    FIELD NAME
   NO.  ROW  COLUMN     OR 'TEXT'

    1    12    45       'JULY 31, 1983'
    2    14     1       FIRSTNAME
    3    14    17       LASTNAME
    4    15     1       ADDRESS1
    5    16     1       ADDRESS2
    6    17     1       CITY
    7    17    32       ZIP
    8    19     1       'DEAR'
    9    19     6       FIRSTNAME
   10    19    21       ':'
   11    21     6       'EVERY YEAR YOUR COMPANY PROVIDES YOU WI
                        TH A STATEMENT OF EMPLOYEE'
   12    22     1       'BENEFITS TO WHICH YOU ARE ENTITLED AS A
                        N EMPLOYEE OF DISTRIBUTING'
   13    23     1       'COMPANY, INC. YOUR CURRENT SALARY IS $'
   14    23    39       SALARY
   15    23    49       '. IN ADDITION, YOU ARE'
   16    25     1       'EARNED'
   17    25     8       CREDITS
```

```
   18     25     11        'CREDITS FOR TAKING COURSES IN THE COMPA
                           NY'S EDUCATIONAL'
   19     26      1        'PROGRAM.'
   20     28     45        'SINCERELY,'
   21     30     45        'DISTRIBUTING COMPANY, INC.'
   22     45      1        'ENTITLED MANY OTHER EMPLOYEE BENEFITS.
                           IN THE PAST YEAR, YOU HAVE'

* LIST, DELETE, ADD, OR CHANGE? (L/D/A/C):
```

As you can see, in form listings the actual text for text items is enclosed in quotation marks. In this way, text can easily be distinguished from field names.

In reviewing this form specification and checking for errors and omissions, you first notice that rows 17 and 19 should have been included as rows from which extra blanks should be trimmed. The field values *CITY* and *ZIP* are to appear on row 17, and excess blanks may appear between these two fields if the value for *CITY* does not fill its entire field width. Similarly, extra spaces may occur between the *FIRSTNAME* value in the salutation and the following colon. The form specification will therefore have to be changed so that rows 17 and 19 are included for trimming of excess blanks. Also, while reading over the items, you notice that the text does not flow properly from item 15 to 16. After reviewing the entire form listing, you realize that the text for item 22 really belongs between items 15 and 16. Looking back to where you were itemizing the text, you can see that the location for this item was entered incorrectly. In addition to changing its position, the text for item 22 needs to be edited so it is grammatically correct.

The following dialogue with MABRA shows how you would make the necessary changes to form specification 2.

```
* LIST, DELETE, ADD, OR CHANGE? (L/D/A/C): C
* CHANGE WHICH SPECIFICATION NUMBER: 2
* SET NEW FORM DIMENSIONS? (Y/N): N
* ADD, DELETE, OR CHANGE AN ITEM? (A/D/C): C
* CHANGE WHICH ITEM NUMBER? 22
* FIELD OR TEXT? (F/T): T
* TEXT: ENTITLED TO MANY OTHER EMPLOYEE BENEFITS. IN THE PAST YEAR, YOU HAVE
* ROW: 24
* COLUMN: 1
* CHANGE WHICH ITEM NUMBER? STOP
* ADD, DELETE, OR CHANGE AN ITEM? (A/D/C): STOP
* REMOVE BLANK ROWS? (Y/N): Y
* CHANGE TRIMMING OF EXTRA BLANKS? (Y/N): Y
* TRIM EXTRA BLANKS FROM WHICH ROWS: 14 17 19 23 25
* OK TO UPDATE? (Y/N): Y
* SPECIFICATION 2 UPDATED.
* LIST, DELETE, ADD, OR CHANGE? (L/D/A/C): STOP
01:
```

When *SF* prompts for another option, you give the response required to change a form specification. After giving the appropriate specification number, you are asked if you want to set new form dimensions. In this case, changing the form dimensions is not required. You do, however, need to change an item. Once you have entered the appropriate item number, you are prompted in the same way that you were when you originally defined the item. So, to change the item, you simply re-specify it—this time entering the correct text and the correct position. Finally, you must re-specify the desired trimming of blank rows and excess blank spaces caused by empty or "short" fields. When you are prompted

to enter the numbers of the rows that require trimming, you must enter the original three rows, as well as rows 17 and 19 which you recently discovered also need trimming.

Having updated specification 2 with the necessary corrections, you can terminate the *SF* command.

Now you want to use the *PF* command to print a report using form specification 2.

```
01:PF
* PRINT FORM
* PRINT WHICH FORM SPECIFICATION: 2
* CONSTRAINTS: 11 17
     2 FOUND.
* SORT FIELDS: LASTNAME,FIRSTNAME
* TERMINAL OR HIGH SPEED PRINT? (T/H): T
* PAUSE AFTER EACH PAGE? (Y/N): Y
* ALIGN PAPER, TYPE SPACE, CR TO START.
```

JULY 31, 1983

DAVID BLACKMAN
71 ANGORA ST.
APT. 13
ROCHESTER, NY 14622

DEAR DAVID:

EVERY YEAR YOUR COMPANY PROVIDES YOU WITH A STATEMENT OF EMPLOYEE BENEFITS TO WHICH YOU ARE ENTITLED AS AN EMPLOYEE OF DISTRIBUTING COMPANY, INC. YOUR CURRENT SALARY IS $ 25,600.00. IN ADDITION, YOU ARE ENTITLED TO MANY OTHER EMPLOYEE BENEFITS. IN THE PAST YEAR, YOU HAVE EARNED 20 CREDITS FOR TAKING COURSES IN THE COMPANY'S EDUCATIONAL PROGRAM.

SINCERELY,

DISTRIBUTING COMPANY, INC.

JULY 31, 1983

ANDREW BOOTH
125 SOUTH UNION ST.
NORFOLK, VA 38748

DEAR ANDREW:

EVERY YEAR YOUR COMPANY PROVIDES YOU WITH A STATEMENT OF EMPLOYEE BENEFITS TO WHICH YOU ARE ENTITLED AS AN EMPLOYEE OF DISTRIBUTING COMPANY, INC. YOUR CURRENT SALARY IS $ 33,400.00. IN ADDITION, YOU ARE

```
ENTITLED TO MANY OTHER EMPLOYEE BENEFITS. IN THE PAST YEAR, YOU HAVE
EARNED 18 CREDITS FOR TAKING COURSES IN THE COMPANY'S EDUCATIONAL
PROGRAM.

                                        SINCERELY,

                                        DISTRIBUTING COMPANY, INC.

* PRINT WHICH FORM SPECIFICATION: STOP
01:
```

As expected, each report (letter) has the same "text", while the field values change with each record that is selected—so each letter is "an original". As per the form specification, the blank row is removed from the inside address of the first letter, but the blank line between the inside address and the salutation is retained. Extra blank spaces have also been trimmed from the appropriate rows of the inside address and around the field values inserted in the body of the letter.

Since you do not intend to use either of the form specifications again, they can be deleted.

```
01:SF
* SPECIFY FORM
* THERE ARE 2 FORM SPECIFICATIONS
* LIST, DELETE, ADD, OR CHANGE? (L/D/A/C): D
* DELETE WHICH SPECIFICATION NUMBERS: 1 2
* OK TO DELETE? (Y/N): Y
* DONE.
```

As you have finished your work with MABRA for the time being, you exit from the MABRA program. And, having no other computer-related work to do, you also sign off the computer.

```
* LIST, DELETE, ADD, OR CHANGE? (L/D/A/C): EXIT
      )OFF
```

Chapter 5

MAINTAINING MABRA SYSTEMS

Maintaining data that is stored in a manual filing system can be a very tedious and time-consuming process, as many people know. Integrating new data into the system, locating and updating data, and disposing of obsolete data can all involve considerable effort on the part of one person or many.

This chapter will illustrate in contrast how easy it is to maintain the data stored in a MABRA system. New data can be added, and existing data can be located, updated, or deleted conveniently and efficiently. All commands related to system maintenance including those required to list, withdraw, and remove updates, copy records from other systems, and define new field orders for adding records are described in this chapter.

Note: To use any of the commands discussed in this chapter, you must be in *DATA MODE*. That is, you must have *ENTER*ed a MABRA system.

Adding Records

To add a record to a MABRA system, you use the *AR* (Add Record(s)) command. When you enter this command, you are prompted to enter a value for each field that has been defined for the system you are using. Field values are prompted for, one at a time, in the order that the fields were originally added to the system. The *FO* command, discussed later in this section, does however allow you to alter this order. Once you have entered values for all existing fields, another prompt for the first field is displayed. This allows you to enter values for a second record. This pattern is repeated until you enter *STOP* in response to one of *AR*'s prompts.

The sample personnel system will continue to be used to demonstrate the commands in this chapter. However, for the examples that illustrate the *AR* and *FO* commands, a condensed version of the system is used. *LASTNAME*, *FIRSTNAME*, *INITIAL*, *BIRTHDATE*, *SEX*, and *SALARY* are the only fields that are defined.

```
01:AR
* ADD RECORD(S)

[1]
LASTNAME  :DUNCAN
FIRSTNAME :KEVIN
INITIAL   :J
BIRTHDATE :140948
SEX       :M
SALARY    :31840

[2]
LASTNAME  :ANDERS
FIRSTNAME :JAMES
INITIAL   :N
BIRTHDATE :071150
SEX       :MALE
? TRUNCATED TO: 'M'.
SALARY    :297W0
? INVALID FIELD TYPE, RE-ENTER
```

```
SALARY     :29700

[3]
LASTNAME   :
```

When you enter date values, they must be entered in the format **ddmmyy**—day, month, and year (unless the *USDATES* state has been set, see Chapter 3)—with no embedded spaces or punctuation. Similarly, numeric field values, such as *SALARY*, must be entered without punctuation and with no accompanying symbols.

If the value that you enter for a field has more elements than the field is defined to have, MABRA truncates (shortens) your entry and discards the extra elements. In the case of character fields such as *LASTNAME* and *SEX*, elements are simply characters or letters. A more complete discussion of elements is given in Chapter 6. In the example above, when "*MALE*" is entered for the field *SEX*, it is truncated to "*M*". This is because *SEX* is defined as a one-element field. MABRA always informs you when it truncates any field values that you enter while adding records.

If you enter a value that has fewer elements than the field definition specifies, the value is padded with the appropriate filler data along the field's right boundary. So when *DUNCAN* was entered as the value for *LASTNAME*, it would then have been padded with fourteen blank spaces (as *LASTNAME* has a field width of 20 characters).

When you enter a value that is invalid for a particular field's data type (as in the case of *SALARY* above), MABRA informs you by displaying a question mark (*?*) followed by a brief error message. You are then re-prompted for that same field again.

While character fields are not validated, MABRA does perform the following checks on the different numeric fields:

INTEGER all elements must be valid integers (whole numbers) from `¯2*31` through `¯1+2*31`.

REAL all elements must be valid numerics (integers and decimals).

BINARY all elements must be 0 or 1 and separated by a blank space.

DATE must be in the form **ddmmyy**, **mmyy**, **yy**, or `0`, where **dd**≤`31`, **mm**≤`12`, **yy**≤`99`, and `0` indicates that there is no date. If the *USDATES* state is set, the form must be **mmddyy**.

```
[3]
LASTNAME   :JACOBS
FIRSTNAME  :JERMY
INITIAL    :O
BIRTHDATE  :271056
SEX        :→FIRSTNAME
FIRSTNAME  :JEREMY
INITIAL    :→SEX
SEX        :M
SALARY     :18400

[4]
LASTNAME   :
```

If, while you are adding a record, you realize that a field value for that record has been entered

incorrectly, you can go back to that field using the "go to" feature. In the example above, the value for *FIRSTNAME* is entered in error. You realize this just as you are prompted for a value for *SEX*. So instead of entering a value for *SEX*, you enter → (branch to) followed by the name of the field you want to back up to, in this case *FIRSTNAME*. When the value for *FIRSTNAME* has been entered again, you are prompted for *INITIAL*, which is the field that normally follows *FIRSTNAME*. Since you already entered a correct value for both *INITIAL* and *BIRTHDATE*, you use "go to" again, this time to skip forward to the next field you want to enter a value for. So "go to" (→) can be used to move backward or forward within a record.

```
[4]
LASTNAME   :JACOBSON;LINDSAY;P;210451;F;24750

[5]
LASTNAME   :DAVIDSON;JENNIFER;S;280954;F;18600

[6]
LASTNAME   :DOUGLAS;EMILY;D;110753;F;14400;TOMPKINS;BRUCE
INITIAL    :B;300146;M;27800

[8]
LASTNAME   :
```

When you know the sequence of prompts that MABRA will be issuing, you can "pre-answer" or "answer-ahead" by typing several responses on one line separated by semi-colons. This is a fast and economical way to use MABRA, as it reduces the amount of time you spend communicating with the computer.

When you are adding records and you answer-ahead, MABRA assigns the values that are entered to the fields in the order that the prompts would have been displayed. Thus *JACOBSON* is assigned to *LASTNAME*, *LINDSAY* is assigned to *FIRSTNAME*, *P* is assigned to *INITIAL*, and so on.

```
[8]
LASTNAME   :THOMSON;GRAHAM;  ;200548;M;35600

[9]
LASTNAME   :MCCARTHY;S;  ;  ;F;41900

[10]
LASTNAME   :KENNEDY;MARY;E;→

[11]
LASTNAME   :NICHOLS;ROBERT;K;170752;M;13700

[12]
LASTNAME   :STOP
* UPDATE REQUEST NO. 5 FILED.
01:
```

When you do not have a value for a field, you can leave it empty by typing a blank space instead of a value. Fields that are left empty for a particular record are given fill values. The fill values are blank spaces for character fields and zeros for numeric fields. In the example above, the record for *GRAHAM THOMSON* will contain fill values (blanks) for the *INITIAL* field, while the record for *S MCCARTHY* (a full first name was not available when the record was added) will contain fill values for both the *INITIAL* and *BIRTHDATE* fields.

When you use "go to" without pointing to a field name, you skip over any fields for the record being added that you have not yet entered values for. You are then prompted for the first field of the next record. By using → by itself, you are telling MABRA two things. First, you want to leave any remaining fields for the current record empty, and second, you want to start entering an entirely new record. Again, any fields that you leave empty are given fill values.

You enter *STOP* to tell MABRA that you no longer want to add any records. You should be aware that when you type *STOP* you also abort the input of the current record. Ordinarily, you type *STOP* in response to the first prompt for a new record, so aborting the current record has no real effect. However, if you type *STOP* after you have started entering values for a new record, those values are ignored.

When you terminate the *AR* command, MABRA files a request to update the system and assigns an identification number to the request. The update request to add the 11 records as entered above is request no. 5.

MABRA does not immediately perform additions, changes, or deletions to a system's records. Rather, each request to add, change, or delete records is filed (i.e., stored) so it can be reviewed if necessary before the changes are made to the system. The *UP* (Update) command is used to actually update the system, and it is discussed later in this chapter.

There are really two conditions under which MABRA files update requests for *AR*. The first, we have learned, is when you enter *STOP* in response to one of *AR*'s prompts. The second condition relates to a limit on the number of records that an update task can process. When you have entered the number of records that equals this limit, MABRA will automatically file an update request. Because the size of one record varies from system to system, the number of records that an update task can accommodate also varies for each MABRA system.

It is wise to plan your use of MABRA in such a way that all your work with one command is done at the same time. This is especially true with the *AR* command. By adding as many records as possible with one update request, you are using MABRA more economically.

When you are adding records, it is often more convenient to enter field values in a different order or to omit certain fields from the prompting sequence. For example, you may want to be prompted for field values in the same order as they are entered on a source document that is being used to supply the data for entry via *AR*. Or you may want to use *AR* for only some fields, adding the remaining field values using other MABRA commands. Being able to omit certain fields from a prompting sequence means that one person could perhaps enter the majority of the fields using *AR*, while any fields containing confidential data (e.g., *SALARY*) could be added later by a more senior employee. For these reasons, the *FO* (Field Order) command allows you to specify alternate field orders for the *AR* prompting sequence.

```
01:FO
* ADD RECORD FIELD ORDER
* THERE IS 1 AR FIELD ORDER
* ADD, CHANGE, DELETE, OR LIST? (A/C/D/L): A
* ADDING FIELD ORDER 2
* FIELDS: FIRSTNAME,INITIAL,LASTNAME,SEX
* MORE? (Y/N): Y
* FIELDS: BIRTHDATE
* MORE? (Y/N): N
* ADDED
* ADD, CHANGE, DELETE, OR LIST? (A/C/D/L):
```

When you enter the *FO* command, you are first told how many field orders already exist. When only one order exists, it is the original order in which you defined the system's fields.

You are then given the choice of adding, changing, deleting, or listing field orders. In this case, a new field order is being added. MABRA assigns an identification number to this field order so it can be referred to when it needs to be changed, deleted, listed, or used by *AR*. The field order added above is assigned the identification number 2.

Next, the names of the fields to be included in the field order must be entered in the same order that you want *AR* to prompt for them. After you type a line of field names, separated by commas, you are asked if there are more fields to be included in the field order. Because some field orders may contain more fields than you can type on one line, you are given the option of entering more fields. In this example, the fields are entered on two lines. When you respond that there are no more fields to be included, the field order is added.

FO then repeats the *ADD, CHANGE, DELETE, OR LIST* prompt. The option of listing field orders is now selected.

```
* ADD, CHANGE, DELETE, OR LIST? (A/C/D/L): L
* LIST WHICH FIELD ORDERS: 1 2
* ALIGN PAPER, TYPE SPACE, CR TO START.

1               2
LASTNAME        FIRSTNAME
FIRSTNAME       INITIAL
INITIAL         LASTNAME
BIRTHDATE       SEX
SEX             BIRTHDATE
SALARY

* ADD, CHANGE, DELETE, OR LIST? (A/C/D/L): STOP
01:
```

When MABRA asks which field orders you want to list, you must enter a list of field order numbers, separated by blanks. You can enter *ALL* if you want all field orders listed.

Above, you request that field orders 1 and 2 both be listed. Field order 1 is the original sequence for adding field values to a record (i.e., the order in which the fields were added to the system), and field order 2 is the sequence just defined. Because field order 2 does not include *SALARY*, *AR* will not prompt for this field when order 2 is being used. For records added using field order 2, *SALARY* will automatically be assigned the fill value for the data type of that field. When you are using a field order that suppresses fields, you cannot use the "jump to" (→) feature to access a suppressed field.

When you have finished with the *FO* command, you enter *STOP* in response to the *ADD, CHANGE, DELETE, OR LIST* prompt.

When more than one field order exists for a MABRA system, *AR* always asks you to specify a field order before it begins prompting for field values.

```
01:AR
* ADD RECORD(S)
* FIELD ORDER: 2

[1]
FIRSTNAME :PAUL
INITIAL   :J
```

```
LASTNAME  :HOPKINS
SEX       :M
BIRTHDATE :090953
[2]
FIRSTNAME :
.
.
.
```

Deleting Records

The *DR* (Delete Record(s)) command is used when you wish to delete one or more records from a MABRA system.

```
01:DR
* DELETE RECORD(S)
* CONSTRAINTS: 5
     1 FOUND.
* PRINT RECORDS TO BE  DELETED? (Y/N): Y
* PRINT FIELDS: LASTNAME,FIRSTNAME,IDNUMBER,Δ
* SORT FIELDS: NONE
* ALIGN PAPER, TYPE SPACE, CR TO START.

LASTNAME              FIRSTNAME       IDNUMBER   Δ

POLONIO               MARK            0705       5

* FILE UPDATE REQUEST? (Y/N): Y
* UPDATE REQUEST NO. 6 FILED.
* CONSTRAINTS: STOP
01:
```

The first thing you are prompted for when you enter the *DR* command is *CONSTRAINTS*. Any valid constraint that identifies the record(s) to be deleted can be entered. In this example, the record to be deleted is identified by specifying its unique record number (Δ number).

You are then given the option of printing the records you will be deleting. Printing records before they are deleted is recommended. It gives you the opportunity to ensure that you are deleting the correct records, and it also provides you with a printed copy of deleted records. If you do request that the records be printed, you must also specify which fields of the records you want displayed and which fields (if any) are to be used as sort fields. Fields are printed in the order in which they are specified. You can also print field derivatives. In this case, the values for *LASTNAME*, *FIRSTNAME*, *IDNUMBER*, and Δ are printed.

When you are prompted to enter a list of *PRINT FIELDS*, entering *ALL* will result in values being printed for all fields that you have added to the system. This does not include values for Δ or Δ*COUNT* which are added to every system by MABRA. So to include these fields in a display, they would have to be specified in addition to *ALL*.

No sorting is required in this example since only one record is to be printed. *NONE* is the appropriate response to the prompt for *SORT FIELDS* when records do not need to be sorted.

After the records have been displayed, you are asked to confirm that the request to delete the records be filed. When the request has been filed, you are again prompted for *CONSTRAINTS*. If you do not want to delete any more records, you terminate the *DR* command by entering *STOP*.

Deleted records are not physically erased from the system. Rather, they are flagged (i.e., redefined) as deleted, and from then on are ignored by MABRA. To actually remove deleted records from the system files, you need to use the *RB* command (described in Chapter 6).

Changing Records

There are two different commands that are used to change field values for selected records. When you have a group of records that all need to have the value for a particular field changed to the *same new value*, you use the *CR* (Change Records) command. For example, if a company decided to combine two departments and form a new department with a new name and number, all employees formerly in either of the old departments would need their records updated to reflect the new department number. The *CR* command would make the required change for all identified records in one step.

On the other hand, when each record in a group needs to have the value for a particular field changed to a *different new value*, you need to use the *CD* (Change Different Values) command. If a certain branch office was closed and the employees from that office were all relocated to different new offices, the records for those employees would need to be updated. The branch office field for each record would need to be changed to a different new value.

If field values need to be changed for only one record, either command can be used. However, *CD* is the command you use most often to make record changes.

The following example illustrates the use of the *CR* command.

```
01:CR
* CHANGE RECORDS
* CONSTRAINTS: SALARY>25000
    62 FOUND.
* PRINT? (Y/N): N
* ENTER CHANGES.
:EXEMPT←1
:STOP
* LIST CHANGES? (Y/N): Y
EXEMPT          ←1
* FILE UPDATE REQUEST? (Y/N): Y
* UPDATE REQUEST NO. 7 FILED.
* CONSTRAINTS: STOP
01:
```

In this example, the constraint *SALARY*>25000 identifies those records that need to be changed. You then proceed to specify the changes without printing the records. If you had asked for the records to be printed, you would have been prompted for *PRINT FIELDS* and *SORT FIELDS*.

Next, you are instructed to enter your changes. MABRA prompts for each change with a colon (:). When you use *CR* to change a field value (in one or many records), the statement for specifying the

change consists of three parts: the field name, the assignment arrow (←), and the field's new value. In this case, the *EXEMPT* field is assigned a new value of 1 in all 62 records identified by the constraint.

When you have finished entering changes, you type *STOP* in response to the prompt for a change (:). You can then list the changes if you wish. If you are satisfied that all changes have been specified correctly, you would instruct MABRA to proceed with filing the update request. Once the request has been filed, you are again prompted for *CONSTRAINTS*. If you have no more changes to make using the *CR* command, you type *STOP* at this point.

The next example shows how records are modified using the *CD* command.

```
01:CD
* CHANGE DIFFERENT VALUES
* CHANGE FIELDS: IDNUMBER,ADDRESS1,ADDRESS2,CITY,ZIP
* MORE? (Y/N): N
* PRINT FIELDS: LASTNAME,FIRSTNAME,IDNUMBER
* SORT FIELDS: LASTNAME
* CONSTRAINTS: 148 154 111
     3 FOUND.

[1] BARRETT              ;ELIZABETH       ;4072

  IDNUMBER: 6355
  ADDRESS1: 1455 BROADWAY AVE.
  ADDRESS2: APT. 530
  CITY    : BURLINGTON, VT
  ZIP     : 23415

[2] DONOVAN              ;VICTOR          ;4521

  IDNUMBER: 7411
  ADDRESS1: 81 CENTRAL PKWY.
  ADDRESS2:
  CITY    : ROCHESTER, NY
  ZIP     : 17351

[3] NELSON               ;ARTHUR          ;4933

  IDNUMBER: 6341
  ADDRESS1: 409 BIRCHMOUNT RD.
  ADDRESS2:
  CITY    : ROCHESTER, NY
  ZIP     : 14866
* UPDATE REQUEST NO. 8 FILED.
* CONSTRAINTS: STOP
* CHANGE FIELDS: STOP
01:
```

As we have learned, the *CR* command is used when you want to change a particular field value for every record in a group to the *same* new value. The function of the *CD* command, however, is to make *different* changes to the field values of each record in a group.

After entering the *CD* command, you are prompted for the names of the fields you want to change. In this example, *IDNUMBER*, *ADDRESS1*, *ADDRESS2*, *CITY*, and *ZIP* are the fields being changed.

In order to identify each record before it is changed, the values for certain fields need to be printed. *LASTNAME* and *FIRSTNAME* are the fields that identify each record in the example above. It is also useful to print the existing values for some or all of the fields you intend to change, so *IDNUMBER* is included in the list of *PRINT FIELDS* as well. *LASTNAME* is the field chosen as the *SORT FIELD*.

Finally, a constraint must be given to identify the records that need their field values updated. Here you enter three Δ record numbers.

For each record, MABRA prints the values for the specified *PRINT FIELDS* and then prompts for the new values of the fields being changed. The same rules for inputting data values discussed in the section, "Adding Records", apply when you are entering new values with the *CD* command.

Once new field values have been entered for all records selected by the constraint, MABRA automatically files an update request.

The "go to" feature works basically the same with *CD* as it does with *AR*, except that with *CD* any fields that are skipped over are not given fill values. Rather those fields retain their current value.

When the update request has been filed, you are given the opportunity to enter another constraint to define another group of records that needs the same change fields updated—in this case, *IDNUMBER*, *ADDRESS1*, *ADDRESS2*, *CITY*, and *ZIP*. Since no other records require new values for these fields, you enter *STOP*. You are then prompted for new *CHANGE FIELDS*, and since you have no more changes to make at this time, you enter *STOP* again to exit from the *CD* command.

When new fields are added to a system that already contains a number of records, the existing records are initially given fill values for those fields. *CR* and *CD* are used in such cases to change those fill values to the appropriate field values.

Listing Update Requests

The reason MABRA does not perform record additions, deletions, or changes immediately after they are specified is to give you an opportunity to check them to ensure that they are correct. Errors are much more difficult to detect once modifications have been made to a MABRA system, so it is advisable to check update requests carefully before they are processed.

The *LU* (List Update Requests) command is used to list update requests.

```
01:LU
* LIST UPDATE REQUESTS
* COMPLETED UPDATE REQUEST NUMBERS: 1 → 4
* PENDING UPDATE REQUEST NUMBERS: 5 → 8
* ENTER UPDATE REQUEST NUMBER RANGE: 6 8
* TERMINAL OR HIGH SPEED PRINT? (T/H): T
* ALIGN PAPER, TYPE SPACE, CR TO START.

* REQ=6, USER=1234567, DATE=09/08/83, TIME=19:11:23, COMMAND=DR
  CONSTRAINTS: 5
       1 FOUND.
* REQ=7, USER=1234567, DATE=09/08/83, TIME=19:12:12, COMMAND=CR
  CONSTRAINTS: SALARY>25000
      62 FOUND.
  EXEMPT ←1
* REQ=8, USER=1234567, DATE=09/08/83, TIME=19:16:17, COMMAND=CD
```

```
CONSTRAINTS: 148 154 111
     3 FOUND.
Δ        :111
IDNUMBER:6341
ADDRESS1:409 BIRCHMOUNT RD.
ADDRESS2:
CITY     :ROCHESTER, NY
ZIP      :14866

Δ        :148
IDNUMBER:6355
ADDRESS1:1455 BROADWAY AVE.
ADDRESS2:APT. 530
CITY     :BURLINGTON, VT
ZIP      :23415

Δ        :154
IDNUMBER:7411
ADDRESS1:81 CENTRAL PKWY.
ADDRESS2:
CITY     :ROCHESTER, NY
ZIP      :17351

* ENTER UPDATE REQUEST NUMBER RANGE:
```

After entering *LU*, completed update request numbers (i.e., those that have been processed) and pending request numbers are listed. You are then asked to specify which update requests you want to have displayed. Any request, completed or pending, can be listed. You can enter either one number to list only one update request, or two numbers which MABRA interprets as a range of numbers. For example, to display update requests 3, 4, 5, and 6, you would enter 3 6.

Update requests can be listed at your terminal, or they can be printed on one of the computer's highspeed printers. Because the requests listed above were short, they were printed at the terminal.

At the beginning of each update request, MABRA prints the following information: the update request number, the account number of the user who filed the update request, the date and time the request was filed, and the command used to file the update request. For requests filed using the *DR*, *CR*, or *CD* commands, the constraint used to identify the records to be deleted or changed is also displayed. To identify records being modified by *CD*, Δ record numbers are printed.

LU is similar to the report commands discussed in Chapter 4 in that it waits for you to press the SPACE bar and the RETURN key after the listing is printed before it issues any further prompts.

Withdrawn update requests (see next section) can also be displayed. If you display a request that has been withdrawn, it will be labelled appropriately.

The next example shows how listings of update requests are directed to a highspeed printer.

```
* ENTER UPDATE REQUEST NUMBER RANGE: 5
* TERMINAL OR HIGH SPEED PRINT? (T/H): H
* PRINT SPECS: NONE
* ENTER DELIVERY INSTRUCTIONS. TYPE 'STOP' TO END INSTRUCTIONS.
DISTRIBUTING COMPANY, INC.
EASTERN DIVISION PERSONNEL
```

```
847 EAST DRIVE
ROCHESTER, NY 14932
ATTN: L. WHITTAKER
STOP
* SUBMIT REQUEST? (Y/N): Y
* PRINT FILE IS: P730053107
* NTASK 3851 RUNNING.
* ENTER UPDATE REQUEST NUMBER RANGE: STOP
01:
```

The option of printing update listings on a highspeed printer is usually preferred when you know the listings will be very lengthy. When you do choose this option, you are asked to provide some additional information. In response to the prompt for *PRINT SPECS*, you would enter any highspeed print specifications required to process the print request (e.g., *DO2*, *NOERASE*). For further information regarding highspeed print specifications, you should refer to the I.P. Sharp publication, *SATN 8; HSPRINT*.

When you are prompted for delivery instructions, you must enter all the information required to get the listing delivered to you—a complete address plus any special mailing instructions that may be necessary. You can use as many lines as you require, but when you have finished entering your instructions, you must type *STOP* on a line by itself.

After you have instructed MABRA to proceed and submit the print request, the name assigned to your temporary print file is displayed. (This is a file that must be created in order to complete the print request.) Your account number becomes part of the file name as well. This file is automatically erased once the printing has completed, unless you enter the specification *NOERASE* in response to the *PRINT SPECS* prompt.

An N-task is started up to process the print request, and the number of the N-task is displayed.

When you have finished listing update requests, you type *STOP* in response to the *ENTER UPDATE REQUEST NUMBER RANGE* prompt.

Withdrawing Update Requests

Any update request that is still pending can be withdrawn if you decide for any reason that it should not be processed. Once an update request is withdrawn, it is ignored by *UP* when update requests are processed.

To withdraw an update request, you use the *WU* command.

```
01:WU
* WITHDRAW UPDATE REQUESTS
* WHICH NUMBER(S)? 6
* DONE.
* WHICH NUMBER(S)? STOP
01:
```

When you execute the *WU* command, you are asked to enter the numbers of the update requests you wish to withdraw. Numbers should be separated by blanks if you enter more than one. Once the update requests have been withdrawn, MABRA displays a confirming message and again prompts for update request numbers. Entering *STOP* in response to this prompt terminates the *WU* command.

The next time you execute the *UP* command (see next section), any withdrawn update requests are still considered pending and are included in the update task. However, they are effectively ignored by *UP* and will not be used to update the system.

When an update request is withdrawn, it is still retained on file to provide an audit trail.

Updating a MABRA System

As we have learned, when record additions, deletions, or changes are specified via the *AR*, *DR*, *CR*, and *CD* commands, MABRA files update requests to perform the required system modifications. To get MABRA to process these requests, you use the *UP* (Update) command.

MABRA uses a separate command to actually process update requests for several reasons. First, by collecting a number of changes into one task, MABRA reduces system costs. If you realize that an error has been made, you can withdraw a particular request before it is processed. Also, by looking at an accumulation of update requests, you have the opportunity to review your work before MABRA actually processes it.

```
01:UP
* INITIATE UPDATE TASK
* 3 REQUESTS PENDING UPDATE
* RUN AS N-TASK OR B-TASK? (N/B): N
* N-TASK NO.3562
01:
```

When you enter the *UP* command, you are first told how many update requests are pending (i.e., waiting for the update task to be run). Then you are asked if the update task should be run as an N-task or a B-task. You will remember many of the report commands discussed in the previous chapter had a similar prompt.

N-tasks and B-tasks are two kinds of jobs that MABRA can run to update the system. Selecting an N-task would mean the update would be processed immediately. You would ordinarily choose an N-task to process updates when you need to work with the updated system on the same day.

If a B-task is selected, the update would be processed when the computer is not very busy. For this reason, B-tasks are usually the more economical choice. You would choose a B-task to process updates when you do not need to work with the updated system until the next day. B-tasks only run after 23:00 U.T.C. (i.e., after 11:00 p.m. Greenwich Mean Time).

It is important to be aware that once a B-task has been submitted for a particular MABRA system, you cannot process an N-task for that same system until the B-task completes.

Once you have chosen the appropriate way to have your update processed, MABRA displays the identification number of either the N-task or the B-task that has been submitted to process your update. When MABRA submits an N-task or a B-task to process any kind of request, the work is taken care of separately from your terminal session. You do not have to wait for the task to complete. Instead, you can do other work or even exit from MABRA and sign off the system. The N-task or B-task will complete on its own. For further information on N-tasks and B-tasks, you can refer to the I.P. Sharp publication, *Batch Tasks in SHARP APL*.

Each time you enter *DATA MODE*, MABRA takes a "snapshot" of the update status of the system you are entering. This snapshot or picture of the system remains even though as you work with the system

MABRA may process an update task that adds or deletes records. You would not see those changes until you enter the system during another MABRA session, re-enter it later in the same session, or use the *NS* command. For example, you may have started an update task to process a request to delete several records. However, these records would still appear in record listings until you re-enter the system or use the *NS* command.

Using the *NS* command is more economical than returning to *SYSTEM MODE* and re-entering a MABRA system. This is because every time you enter a system, MABRA must do a number of other things to enable you to use the system besides take a snapshot.

The use of the *NS* command is illustrated in the following example.

```
01:NS
* NEW SNAPSHOT.
* LAST UPDATE:  9/08/83 21:38:06; CPU USED: 8
* NUMBER OF COMPLETED UPDATES: 8
01:
```

NS takes a new snapshot of the system and displays the date and time that the last update task completed as well as the number of completed updates that are now stored.

If an update task is actually running when you enter *NS*, MABRA displays a message informing you that an update task is in progress and then waits until the update task has completed before taking a new snapshot. If the update task takes more than a few seconds, MABRA repeats the "update in progress" message from time to time.

```
01:NS
* NEW SNAPSHOT.
* UPDATE TASK 3562 IN PROGRESS. . .
* UPDATE TASK 3562 IN PROGRESS. . .
* LAST UPDATE:  9/08/83 21:38:06; CPU USED: 8
* NUMBER OF COMPLETED UPDATES: 8
01:
```

If you do not want to wait for the update task to complete, you can press the BREAK or ATTENTION key on your terminal and MABRA will abort the *NS* command and return to the *DATA MODE* prompt. You are then free to do other work at your terminal while the update task runs to completion.

Removing Update Requests

To economize on file storage, you should periodically remove completed update requests. Every time you use *AR*, *DR*, *CR*, and *CD*, an update request is stored on file. The *RC* (Remove Completed Update Requests) command allows you to delete completed update requests from that file.

```
01:RC
* REMOVE COMPLETED UPDATE REQUESTS.
* THERE ARE 8 COMPLETED UPDATE REQUESTS: REQ. NOS. 1 TO 8.
* REMOVE UP TO WHICH REQUEST NUMBER? 7
* DONE.
01:STOP
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

After entering *RC*, you are told how many completed update requests exist. The identification numbers of these completed requests are also displayed. MABRA then removes all completed update requests up to and including the one that you specify in response to the prompt *REMOVE UP TO WHICH REQUEST NUMBER*.

In this example, completed update requests 1 through 7 inclusive are removed. Update request 8, however, would not be removed. There are two reasons for retaining recent update requests even though they have been processed by *UP*. First, you may have a need to list recently completed requests using *LU*. Second, if problems should ever occur with a MABRA system, you can always retrieve a previous version of the system from the system backup tapes and bring it up to date if all update requests that have been processed subsequent to that version of the system are still on file.

Copying Records From Other MABRA Systems

Suppose you have just created a new MABRA system (system 2) called *NORTH AMERICAN PERSONNEL*, and you want to copy certain records from system 1 (*PERSONNEL*) into it. The *CS* (Copy System) command allows you to perform this function.

CS is also useful when you want to set up a system to be used for training or demonstration purposes, or to merge two or more compatible systems.

The *CS* command allows you to copy an entire system, or a part of one.

To use *CS*, you must have entered the system that the records are to be copied into. It is referred to as the **target system**. The system that you want to copy from is called the **source system**.

```
* SYSTEM MODE
:ENTER;2
* DATA MODE, SYSTEM IS 02: NORTH AMERICAN PERSONNEL
02:CS
* COPY SYSTEM
* ENTER SOURCE CREATOR AND SYSTEM NUMBER: 1
* ENTER RECORDS TO BE COPIED IN THE FORM OF A CONSTRAINT
* CONSTRAINTS: IDNUMBER=5
    38 FOUND.
* ENTER FIELDS TO BE COPIED
* FIELDS: ALL
* MORE FIELDS? (Y/N): N
* DO YOU WISH TO OVERWRITE THE TARGET SYSTEM? (Y/N): Y
* ARE YOU SURE? (Y/N): Y
* ENTER MAXIMUM NUMBER OF RECORDS EXPECTED (AT LEAST 38): 200
* SYSTEM COPIED
02:
```

The first thing *CS* prompts you for is the creator and number of the source system (the system you want to copy from). If *CREATOR* is currently set to the account number of the owner of the system you want to copy from, only the system number needs to be entered. You can only copy from a system belonging to another account if you have list access to that system (see Chapter 6 for a description of the different types of access).

You then have to enter a constraint that identifies the records to be copied. Because the first element

of *IDNUMBER* represents the plant in which an employee works, the constraint *IDNUMBER*=5 selects the records of all employees in plant 5. (Remember that *IDNUMBER* is a character field, not a numeric field. Therefore, the constraint *IDNUMBER*=5 selects all records whose *IDNUMBER* field begins with the character 5.)

Next, you have to tell MABRA which fields you want to copy by entering a list of field names, separated by commas. In this case, *ALL* fields are to be copied. Your list of fields does not have to be entered entirely on one line, as MABRA continues to prompt for *MORE FIELDS* until you indicate that you have no more to enter.

You are then asked if you want to overwrite the target system. If you overwrite the target system, any records and fields that currently exist in the target system are discarded. They are replaced by field definitions and records from the source system. If you do not overwrite the target system, you are merely appending new records to the target system.

In the example above, system 2 was created just prior to executing *CS*, so it has no field definitions or records. For this reason, you want to copy not only records from the source system, but also field definitions.

If you do instruct MABRA to overwrite the target system, you are asked to enter a new value for the maximum number of records expected for the system. It must be a number greater than or equal to the number of records you are copying.

MABRA displays a confirming message when the system has been copied.

Later, you decide that the records from system 1 for all employees in plant number 6 should also be copied into system 2.

```
02:CS
* COPY SYSTEM
* ENTER SOURCE CREATOR AND SYSTEM NUMBER: 1
* ENTER RECORDS TO BE COPIED IN THE FORM OF A CONSTRAINT
* CONSTRAINTS: IDNUMBER=6
    24 FOUND.
* ENTER FIELDS TO BE COPIED
* FIELDS: ALL
* MORE FIELDS? (Y/N): N
* DO YOU WISH TO OVERWRITE THE TARGET SYSTEM? (Y/N): N
* ARE YOU SURE? (Y/N): Y
* 24 RECORDS APPENDED TO THE CURRENT SYSTEM
02:
```

This example is very similar to the previous one, except that instead of overwriting the target system, the records from the source system are simply appended to the records that already exist in the target system. It is important to be aware that *CS* can only be used to append records to a system when all the attributes of the fields you are copying are identical in the target and source systems. In the example above, you know the field definitions in the target system (system 2) are identical to the field definitions in the source system (system 1) because the definitions were copied from system 1 in the previous *CS* example.

Chapter 6
SETTING UP AND MANAGING MABRA SYSTEMS

To set up a MABRA system, you first have to create a system, and then *ENTER* that system and define the fields. The commands required to perform these functions will be discussed in this chapter. Other activities involved in setting up a system, such as defining access for other users, are covered here as well.

Then, once a MABRA system has been set up, there are a number of "housekeeping" or management functions that may need to be performed from time to time. For example, a system may need to be resized, renamed, or reblocked, or fields may need to be changed. The commands that allow you to carry out these functions are also discussed in this chapter.

Creating a MABRA System

The following example shows how a new MABRA system is created. The command that is used is *CREATE*, and it is one of the MABRA commands that must be entered in *SYSTEM MODE*.

```
:CREATE
* CREATE SYSTEM
* ENTER SYSTEM NUMBER: 7
* ENTER NAME FOR SYSTEM 07: EMPLOYEE VACATION SYSTEM
* ENTER EXPECTED MAXIMUM NUMBER OF RECORDS: 1000
* SYSTEM 07: EMPLOYEE VACATION SYSTEM
* CREATE? (Y/N): Y
* SYSTEM CREATED.
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

When you invoke the *CREATE* command, you are first prompted for a system number. Every system you create with MABRA must be given a number from 1 to 99. In Chapter 3, it was explained that every MABRA system belonging to a particular account must have a unique number. Therefore, the number you choose for a new system must be different from the numbers of existing MABRA systems that you have created. The system created above was assigned the number 7.

Next, you must enter a name for the system, and it can be up to 40 characters in length. In this case, *EMPLOYEE VACATION SYSTEM* was the name chosen. However, if you should decide at any time that the name is no longer appropriate, it can be changed with the *RENAME* command (described later in this chapter).

MABRA also needs to know how many records you expect the system to have. (The maximum number of records that can be stored in one MABRA system depends on the workspace size. As a guideline, at least 150,000 records can be stored.) In the example above, you entered 1000 because you know that each year one or more records will be added to the system for each employee (and the company currently employs 196 people).

The *RS* (Resize System) command can be used to increase the size of your system if need be. But, to avoid using this command too frequently, it is a good idea to set the maximum number of records generously. It is more economical to set the system size somewhat larger than you anticipate, rather than continually resizing the system.

Finally, MABRA repeats the system number and name so you can check that they are indeed the number and name you intended. You are then asked if the system should be created. If you tell MABRA to proceed, the files required for a new MABRA system are created and initialized, and a confirming message is displayed.

When MABRA creates a system on your account number, you are its **owner**, and no one else can use it unless you give them access. Permitting other users to access your system is discussed later in this chapter.

Adding Fields

Before you can add any records to a new system, you must define fields for that system. In MABRA terms, there are two major types of fields—real fields and pseudo fields. Real fields contain values that are actually stored in a file, while pseudo fields contain values that are computed from other field values rather than stored. All real fields are defined using the *AF* (Add Fields) command, while *AP* (Add Pseudo Fields) is the command used to define pseudo fields. To use either of these commands, you must have *ENTER*ed the system that you are adding the fields to.

In the following examples, four fields are added to the new MABRA system, *EMPLOYEE VACATION SYSTEM*, using the *AF* command.

```
:ENTER;7
* DATA MODE, SYSTEM IS 07: EMPLOYEE VACATION SYSTEM
07:AF
* ADD FIELD
* ENTER FIELD NAME: EMPLOYEE
* TYPE? (CHAR/INTE/REAL/BINA/DATE): CHAR
* NUMBER OF ELEMENTS IN FIELD: 4
* USE DEFAULT OUTPUT FORMAT? (Y/N): Y
* OK TO ADD? (Y/N): Y
* UPDATING FILES.....
* FIELD ADDED.
* ENTER FIELD NAME:
```

Each time an employee takes any vacation time, a record will be added to the system showing the following information: starting date of the vacation, duration (i.e., number of working days missed), and pay status (vacation with pay or without pay). The *EMPLOYEE* field created above will contain the ID number of the employee who has taken the vacation time.

The name you choose for a field must conform to the following restrictions:

- It must contain only letters of the alphabet, the special character Δ, or numbers, and the first character must be a letter. A field name cannot contain any blank spaces or any special symbols.
- It must not exceed 14 characters in length.
- It must be unique. You cannot use the same field name twice in the same system, although the same name can be used in different systems.
- It cannot be *ALL*, *HELP*, *PREVIOUS*, *EXIT*, or *STOP*. MABRA already has special uses for these words.

In addition, to save on keystrokes during MABRA sessions, you should try to choose field names that are short or abbreviated (e.g., *DEPT* rather than *DEPARTMENT*) but still descriptive.

After you have entered a field name, MABRA asks you what type of data is to be stored in the field. There are two basic kinds of data in a MABRA system—character data and numeric data. However, there are four different kinds of numeric data—integer, real, binary, and date. So when you are prompted for a data type, you must select one of the following:

CHAR Character data can contain any of the 26 letters of the English alphabet, special characters (e.g., ≠, ≤, /), the numbers 0 through 9, or blank spaces. Fields that contain names or addresses would be character fields.

INTE Integer fields can store whole numbers ranging from ¯2147483648 to 2147483647. Most numbers that you store in MABRA systems (e.g., monetary amounts, quantities) are stored in integer fields.

REAL Real fields can hold decimal numbers and very small or very large numbers. They are used most often in scientific and engineering applications, but are also used to store monetary values.

BINA Binary fields can contain only the values 1 and 0.

DATE Date fields contain calendar dates stored numerically in the form **ddmmyy**, where the first two digits (dd) represent the day of the month, the next two digits represent the month, and the last two digits represent the year. For example, if a date field contained the value 170549, it would mean the 17th day of May in 1949. A date field can also consist of only four digits or just two digits. If a value consists of only four digits, MABRA interprets it as a month and year (mmyy). Thus 1180 would mean November 1980. Two digits would simply represent a year, so the date value 55 would mean the year 1955.

It should be noted that, unlike most attributes of a field, the data type cannot be changed easily *after* records have been added to the system. If you wish to change a field's data type *before* you add records to the system, you can simply delete the field (using the *DF* command described later in this chapter) and add it again (using *AF*). Most field attributes can be changed using the *CF* command.

The data type you choose affects the amount of storage (i.e., file space) the field will use, and it also affects how efficiently MABRA can search for particular field values. The same amount of storage space in the computer can store the various data types in the following proportions:

Data Type	Number of Elements
Binary	64
Character	8
Integer	2
Real	1
Date	2

This means that you can store 64 binary elements in the same amount of space as it takes to store 8 characters, 2 integers, 2 dates, or 1 real (floating point) number. The fastest searches can usually be done on fields that are binary, whereas the slowest searches are usually done on fields containing real data.

The number of elements in a field (i.e., its width) also affects the field's file storage requirements and the efficiency of searching for a particular value. The fewer the elements in a field, the quicker MABRA can search. You determine a field's width as follows:

Character data:
Each letter, special character, number, or blank is one element.

A	has 1 element
JOHNSON	has 7 elements
1134	has 4 elements
B3A4/G7	has 7 elements
115 *GREEN STREET*	has 16 elements

Integer, real, or binary data:
Each number is one element, regardless of how many digits it takes to print it.

5	has 1 element
1134	has 1 element
6 32 91	has 3 elements
47.358	has 1 element

Date Data:
Date values are *always* one element.

Numbers can be stored in MABRA either in character fields or in any type of numeric field (e.g., integer). A number such as 1134 could be stored in a character field that is at least four elements in width, or in an integer field that is one element in width. In a character field, a character representation of the number is stored. For this reason, calculations cannot be performed on numbers stored in character fields. If you wish to perform any type of calculation, the numbers involved must be stored in numeric fields. Also, some of the powerful output formatting tools (□*FMT* operators) can only be used with numeric data. On the other hand, if you do not need to perform calculations or use any special formatting tools, it is sometimes more economical to use a character field to store numbers. For example, if you need to store a one-digit number (0 through 9) and you don't need to perform arithmetic operations on the data or use special formatting, your storage and searching will be four times more efficient if you store the number in a one-element character field rather than in an integer field.

With this in mind, you specified the character data type when the *EMPLOYEE* field was added above. And a field width of four elements was specified for this field, as all employee identification numbers are four-digit numbers.

When a field is added, MABRA also needs to know if the default output format is appropriate or if you want to enter an alternate format statement. Format statements tell MABRA how you want field values to appear when they are printed in listings and reports. A format statement contains information such as the number of columns used to print a field's values or the number of decimal places to be printed.

The default output formats for the different data types are as follows:

Data Type	Output Format
Character	**w***A*1
Integer	**w***I*10
Real	**w***E*10.5
Binary	**w***I*1
Date	**w***G*□ZZ/ZZ/Z9□

where **w** is the repetition factor for □*FMT*. It would be equal to the value that you specified for the field width.

You will almost always select the default output format for character fields. For numeric field types, however, you may want to devise your own formatting statement. The following section explains how to do this. In the format statements shown below, **q** represents the number of elements in the field; **w** represents the number of columns needed to print the entire formatted value (including a decimal point (.), a negative number sign (¯), or any other symbol); and **d** represents the number of decimal places to be printed, if applicable. Examples of formatted field values are shown in boxes below that show the entire field width.

q*I***w** prints a numeric value with no decimal point.

1*I*10 prints a one-element numeric value using 10 columns. This is the default output format for integer fields. An example of an integer value formatted in this way is:

```
    483918
```

2*I*5 prints a two-element numeric value, using five columns for each element. An example of such a formatted value is:

```
 3461   37
```

1*I*1 prints a one-element numeric field in one column. This is the default output format for binary fields. An example of such a formatted value is:

```
1
```

q*F***w.d** prints a field value that includes a decimal point.

1*F*10.2 prints a one-element numeric field, including a decimal point followed by two decimal places, in 10 columns. An example of a field value formatted in this way is:

```
  71539.00
```

2*F*5.1 prints a two-element numeric field using five columns for each element. Each element is to have a decimal point followed by one decimal place. An example of a field value formatted in this way is:

```
344.0 12.0
```

You can precede the *I* and *F* format statements with one or more of the following **decorators** which further define the appearance of printed output.

B causes a zero (0) value to print as blanks instead of printing 0.

C causes commas to print every three places to the left of the decimal point.

Z adds leading zeros to the left of numeric values, thereby filling the entire width of the field as defined by the format statement.

*K***n** causes a numeric value to be scaled (multiplied by a power of 10) before printing. **n** represents a positive or negative power of 10.

When defining your own format statement, you must make it compatible with the *number of elements* you have defined for the field. If the number of elements in the field and the format statement are incompatible, MABRA rejects your statement as invalid. On the other hand, MABRA cannot validate the *width* you have provided for a field value (or for each element of a field value, if the value contains more than one element). If you do not provide sufficient width for a field value (that is, the number overflows the number of print columns you have specified in your format statement), MABRA prints stars (*) filling the width provided for that value. In particular, you must allow sufficient width for negative signs if anticipated, decimal points in *F* format statements, and commas when using the *C* decorator.

The following table shows how various field values would appear when output according to different format statements.

Number of Elements in Field	Format Statement	Field's Input Value (at *AR*)	Field's Printed Output
1	*BI*10	6526918	6526918
		0	
3	3*BI*3	1	1
		67 948 11	67948 11
		221 0 643	221 643
1	*CF*10.2	600693	600,693.00
		0	0.00
		4829837	**********
2	2*F*5.1	23 56	23.0 56.0
		0	0.0 0.0
		1	1.0 0.0
1	*K*¯2*F*10.2	71150	711.50
		0	0.00
		1	0.01
1	*BZK*¯2*F*10.2	17165	0000171.65
		0	
		1	0000000.01
1	*K*¯1*F*5.1	70	7.0
		1	0.1
		3141	314.1

Notice that the numbers 67 and 948 run together when they are formatted with the 3*BI*3 statement. With multiple element numeric fields, you must be sure to allow sufficient width in your format statement to prevent the numerals of different elements from running together when they are printed. Also, notice that the attempt to format the value 4829837 with the statement *CF*10.2 results in stars

filling the entire format width. This is because insufficient width was provided to print the field value with two decimal places, a decimal point, and commas.

In comparing the two format statements `1F10.2` and `1K¯2F10.2`, you will see that `1F10.2` adds two decimal places at the end but does not change the value of the number. `1K¯210.2`, on the other hand, scales the number, multiplying it by ten to the power `¯2` (the same as dividing by `100`).

These are only a few of the kinds of flexible and powerful format statements that you can enter in response to MABRA's prompt. Format statements give you great versatility when you are planning the layout and organization of printed listings and reports. For more information regarding format statements, refer to the publications, *SHARP APL Reference Manual* and *SHARP APL Report Formatting*.

After you have responded to the prompts regarding output format, MABRA asks if it can add the field. If you are satisfied that you have defined the field's characteristics correctly, you tell MABRA to proceed, and a confirming message is displayed when MABRA has successfully added the field. Then the prompt for a field name is displayed again, giving you the opportunity to add another field.

Three other fields are now added to the *EMPLOYEE VACATION SYSTEM*. The first field is a date field, the second an integer field, and the third field is a binary field.

```
* ENTER FIELD NAME: STARTDATE
* TYPE? (CHAR/INTE/REAL/BINA/DATE): DATE
* USE DEFAULT OUTPUT FORMAT? (Y/N): Y
* OK TO ADD? (Y/N): Y
* UPDATING FILES.....
* FIELD ADDED.
* ENTER FIELD NAME: DURATION
* TYPE? (CHAR/INTE/REAL/BINA/DATE): INTE
* NUMBER OF ELEMENTS IN FIELD: 1
* USE DEFAULT OUTPUT FORMAT? (Y/N): Y
* OK TO ADD? (Y/N): Y
* UPDATING FILES.....
* FIELD ADDED.
* ENTER FIELD NAME: PAYSTATUS
* TYPE? (CHAR/INTE/REAL/BINA/DATE): BINA
* NUMBER OF ELEMENTS IN FIELD: 1
* USE DEFAULT OUTPUT FORMAT? (Y/N): Y
* OK TO ADD? (Y/N): Y
* UPDATING FILES.....
* FIELD ADDED.
* ENTER FIELD NAME: STOP
07:STOP
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

STARTDATE will contain the date that the employee began his or her vacation period. Notice that, when a date field is added, there is no prompt for *NUMBER OF ELEMENTS*. This is because date values are always only one element.

The length of the vacation period, in working days, will be stored in the field *DURATION*. *INTE* was the data type selected for *DURATION*, as certain calculations will need to be performed using the values that will be stored in this field.

PAYSTATUS is a binary field that will contain either the value 0 or 1. The value 1 represents *paid* vacation time, while 0 represents vacation time *without pay*. Binary fields are normally used when you have a condition (in this case, vacation time with pay) that can be either true or false.

When you have no more fields to add, you type *STOP* in response to the *ENTER FIELD NAME* prompt.

Fields can be added to a MABRA system at any time, even if the system already contains a number of records. Any records already in the system when new fields are added will be given appropriate fill values for those fields (blank spaces for character fields, zeros for numeric fields). These records can then be given values for the new fields using either the *CR* (Change Records) or *CD* (Change Different Values) commands.

In the following example, two additional fields are added to system 1.

```
:ENTER;1
* LAST UPDATE:  9/08/83 21:38:06; CPU USED: 8
* NUMBER OF COMPLETED UPDATES: 1
* DATA MODE, SYSTEM IS 01: PERSONNEL
01:AF
* ADD FIELD
* ENTER FIELD NAME: HIREDATE
* TYPE? (CHAR/INTE/REAL/BINA/DATE): DATE
* USE DEFAULT OUTPUT FORMAT? (Y/N): Y
* OK TO ADD? (Y/N): Y
* UPDATING FILES.....
* FIELD ADDED.
* ENTER FIELD NAME: DEPT
* TYPE? (CHAR/INTE/REAL/BINA/DATE): CHAR
* NUMBER OF ELEMENTS IN FIELD: 3
* "SE DEFAULT OUTPUT FORMAT? (Y/N): Y
* OK TO ADD? (Y/N): Y
* UPDATING FILES.....
* FIELD ADDED.
* ENTER FIELD NAME: STOP
01:
```

HIREDATE will contain the date that an employee was hired.

The department that an employee currently works in will be stored in the new field *DEPT*. Although the values stored in *DEPT* will be three-digit numbers, the character (*CHAR*) data type was chosen for this field rather than a numeric field type. No calculations will be performed using these field values, and no special formatting is required.

Pseudo fields contain values that are not stored but are computed using other field values. When the values for a field can be computed using other field values, you can realize a significant savings in terms of storage as well as data entry time. The values are not stored, so you save storage costs, and MABRA computes the values so they do not need to be entered when records are added.

The *AP* (Add Pseudo Field) command is used to add a pseudo field, and its use is illustrated in the following example.

```
01:AP
* ADD PSEUDO FIELD
```

```
* ENTER FIELD NAME: MONTHLYPAY
* TYPE? (CHAR/INTE/REAL/BINA/DATE): INTE
* NUMBER OF ELEMENTS IN FIELD: 1
* USE DEFAULT OUTPUT FORMAT? (Y/N): Y
* ENTER EXPRESSION DEFINING FIELD
:SALARY÷12
* RETAIN THIS FIELD? (Y/N): Y
* OK TO ADD FIELD? (Y/N): Y
* PSEUDO FIELD ADDED.
* ENTER FIELD NAME: STOP
01:
```

As with the *AF* command, *AP* defines a field by its name, data type, width, and output format. You would respond to these initial prompts in the same way that you would if you were adding a real field with *AF*.

What makes a pseudo field different from a real field is the pseudo field's need for a calculation that produces field values based on other field values. When you are prompted for an expression that defines the field, you must enter a valid APL expression. The calculation entered above derives the values for *MONTHLYPAY* by dividing the values for the real field, *SALARY*, by the number of months in a year. This expression is stored. Then, any time you refer to *MONTHLYPAY*, its values are calculated by dividing the values in *SALARY* by 12. If you change any *SALARY* values, those changes will affect the corresponding values of *MONTHLYPAY* automatically.

Finally, MABRA needs to know whether or not the pseudo field is to be retained. Retained pseudo fields stay in the MABRA system for future use. Other users of the system can therefore be given permission to use retained pseudo fields (a discussion of defining access to MABRA systems is found later is this chapter). Session pseudo fields, on the other hand, disappear when you exit from the MABRA program, so they cannot be shared with other users. Only the creator or owner of a MABRA system can add retained pseudo fields. Other users can be given permission to add pseudo fields, but they can only add session pseudo fields.

When you have no more pseudo fields to add, you enter *STOP* in response to the *ENTER FIELD NAME* prompt.

Deleting Fields

If you decide that a field is no longer needed, you can use the *DF* (Delete Field(s)) command to delete it. *DF* is used to delete both real and pseudo fields. In the next example, the field *TERMDATE* is deleted from system 1.

```
01:DF
* DELETE FIELD(S)
* FIELDS TO BE DELETED: TERMDATE
* DELETE TERMDATE? (Y/N): Y
* UPDATING FILES.....
* DELETED.
* FIELDS TO BE DELETED: STOP
01:
```

When you enter *DF*, you are prompted for the name(s) of the field(s) to be deleted. If you are deleting more than one field, the field names must be separated by commas. Once you have told MABRA to

proceed with the deletion, a confirming message is displayed, and you are prompted again for the names of fields to be deleted.

Changing Fields

In the course of working with a MABRA system, you may decide that certain changes need to be made to existing fields. Changes might be required to improve search efficiency, to accommodate data that cannot be added according to current field definitions, or to produce more informative reports.

You use the *CF* (Change Field) command to change field attributes, and it can be used to make changes to either pseudo or real fields. The following examples show how *CF* is used to make a number of different changes to fields in system 1.

```
01:CF
* CHANGE FIELD
* FIELD NAME: MONTHLYPAY
* CHANGE FIELD=MONTHLYPAY
* CHANGE NAME,WIDTH,CODE,FMT,CALC,CLASS: NAME
* ENTER NEW FIELD NAME: GROSSPAY
* OK TO CHANGE NAME FROM 'MONTHLYPAY' TO 'GROSSPAY'? (Y/N): Y
* FIELD NAME CHANGED FROM 'MONTHLYPAY' TO 'GROSSPAY'.
* CHANGE FIELD=GROSSPAY
* CHANGE NAME,WIDTH,CODE,FMT,CALC,CLASS: CALC
* CURRENT CALCULATION IS: ΔSALARY DB 12
* ENTER NEW CALCULATION
:100×SALARY÷12
* OK TO CHANGE CALCULATION? (Y/N): Y
* DONE.
* CHANGE FIELD=GROSSPAY
* CHANGE NAME,WIDTH,CODE,FMT,CALC,CLASS: FMT
* CURRENT FORMAT STATEMENT IS: 1I10
* NEW FORMAT STATEMENT: 1K¯2F10.2
* FORMAT CHANGED.
* CHANGE FIELD=GROSSPAY
* CHANGE NAME,WIDTH,CODE,FMT,CALC,CLASS: STOP
* FIELD NAME:
```

In this interaction with MABRA, three different changes were made to the pseudo field *MONTHLYPAY*. First, the name was changed from *MONTHLYPAY* to *GROSSPAY*. Then a new expression for calculating the field values was entered (so the values for *GROSSPAY* are computed in cents rather than dollars. Finally, the format statement was changed. The new format statement tells MABRA to use 10 columns to print the values for *GROSSPAY* and to print two decimal places (*F*10.2). Furthermore, the value is to be scaled when printed, multiplying the value by 10*¯2 (same as dividing by 100). So the value 14833, for example, would print as 143.33.

The first thing you are prompted for when you enter the *CF* command is the name of the field you want to change. If the name you enter is a valid field name for the current system, MABRA displays the name of the field being changed and then asks you which field attribute you want to alter. A list of the attributes that can be changed is displayed as part of the prompt, and the list is different for pseudo fields and real fields. You can see from the example above that the attributes you can change for a pseudo field are: *NAME*, *WIDTH*, *CODE*, *FMT*, *CALC*, and *CLASS*. Complete descriptions of field attributes are given in the "Adding Fields" section earlier in this chapter.

When you have specified which attribute you want to change, MABRA prompts for the details related to the change. For example, if you are changing the attribute *NAME*, you are asked to enter a new field name. Similarly, you are prompted to enter a new calculation or expression if you are changing *CALC*, and a new format statement if you are changing *FMT*.

When MABRA has made the required change, the name of the field being changed is repeated, and you are again prompted to select an attribute. Once you have made all the necessary changes to a field, you enter *STOP* in response to this prompt. MABRA then returns to the prompt *FIELD NAME*, allowing you to make changes to another field.

In the next example, the real field, *FIRSTNAME*, is changed.

```
* FIELD NAME: FIRSTNAME
* CHANGE FIELD=FIRSTNAME
* CHANGE NAME,WIDTH,INVERT,CODE,FMT: WIDTH
* WIDTH IS CURRENTLY: 15
* ENTER NEW WIDTH: 20
* YOU WISH TO CHANGE WIDTH FROM 15 TO 20? (Y/N): Y
* USE DEFAULT FORMAT? (Y/N): Y
* FORMAT CHANGED.
* WIDTH CHANGED.
* CHANGE FIELD=FIRSTNAME
* CHANGE NAME,WIDTH,INVERT,CODE,FMT: STOP
* FIELD NAME:
```

The field attributes that can be changed for a real field are: *NAME*, *WIDTH*, *INVERT*, *CODE*, and *FMT*. In this case, the *WIDTH* of *FIRSTNAME* was increased from 15 to 20 elements. Increasing the field width means MABRA pads all existing field values with filler data (blanks or zeros) along the right boundary. If you were to decrease the width of a field, all field values would be truncated along the field's new right boundary.

Changing a field's width also requires changing the format statement. Therefore, after MABRA displays the width change you have specified (and you have confirmed that it is correct), you are asked if the default format statement should be used. If you answer *Y*, the format statement and width are changed, and you are asked to select another attribute. If your response is *N*, you must enter a new format statement and then the changes are made.

Next, the *IDNUMBER* and *GROSSPAY* fields are changed.

```
* FIELD NAME: IDNUMBER
* CHANGE FIELD=IDNUMBER
* CHANGE NAME,WIDTH,INVERT,CODE,FMT: INVERT
* FIELD IS ALREADY INVERTED.
* UN-INVERT? (Y/N): Y
* UN-INVERTED.
* CHANGE FIELD=IDNUMBER
* CHANGE NAME,WIDTH,INVERT,CODE,FMT: STOP
* FIELD NAME: GROSSPAY
* CHANGE FIELD=GROSSPAY
* CHANGE NAME,WIDTH,CODE,FMT,CALC,CLASS: CLASS
* CHANGE FIELD TO A REAL FIELD? (Y/N): Y
* UPDATING MASTER FILE. . .
* CHANGE FIELD=GROSSPAY
* CHANGE NAME,WIDTH,INVERT,CODE,FMT: STOP
```

```
* FIELD NAME: STOP
01:STOP
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

Fields that are inverted can be searched more quickly by MABRA. You should, however, decide carefully which fields you will invert. MABRA stores inverted field values *twice* (the second time in a file of inverted field values), so you should be sure that the efficiency gained in searching is justified by the extra storage and the extra cost of updating records in two places. In other words, you should only invert a field that you intend to search frequently. Your system should also contain at least 200 records before you consider inverting fields.

In the above example, *IDNUMBER* is changed from an inverted field to a non-inverted field. You can also invert a non-inverted field using *CF*.

After further consideration of the field *GROSSPAY*, you decide that it should be a real field rather than a pseudo field. So you tell MABRA that you wish to change the field's *CLASS*. Note that while it is possible to change a pseudo field to a real field using *CF*, it is not possible to change a real field to a pseudo field. Should you ever want to make this latter change, you would first have to delete the real field (using the *DF* command) and then add it as a pseudo field (using *AP*).

When you change a pseudo field to a real field, MABRA takes the pseudo field's current values and stores them with the other real field values. From then on, because the field's values are no longer changed automatically when the related field values are changed, you must add and update the values separately using the *AR*, *CR*, and *CD* commands.

When you wish to exit from the *CF* command, you enter *STOP* in response to the *FIELD NAME* prompt.

The last change to be made using *CF* involves a field in the newly-created *EMPLOYEE VACATION SYSTEM*. *PAYSTATUS* is a binary field in that system, so it will contain either the value 0 or 1. When the value of *PAYSTATUS* is 1, it means the employee is taking *paid* vacation time. If the *PAYSTATUS* value is 0, the employee is taking *unpaid* vacation time. 1 and 0, therefore, serve as codes representing the conditions "paid" and "unpaid".

In the next example, *PAYSTATUS* is changed from an un-coded to a coded field. Once a field is defined as coded, you can store descriptions of the different codes (0 and 1, in this case) and have the descriptions included in reports, if you wish. *CF* is also used to change a coded field to an uncoded field.

```
:ENTER;7
* DATA MODE, SYSTEM IS 07: EMPLOYEE VACATION SYSTEM
07:CF
* CHANGE FIELD
* FIELD NAME: PAYSTATUS
* CHANGE FIELD=PAYSTATUS
* CHANGE NAME,WIDTH,INVERT,CODE,FMT: CODE
* FIELD'S CODE DEFINITION IS 'UN-CODED'.
* CHANGE TO 'CODED'? (Y/N): Y
* WIDTH OF CODES' DESCRIPTIONS: 6
* SUGGESTED MAXIMUM NUMBER OF CODES IS: 1307
* IS 1307 OK? (Y/N): Y
* CODED
```

```
* CHANGE FIELD=PAYSTATUS
* CHANGE NAME,WIDTH,INVERT,CODE,FMT: BACK
07:
```

You can code any field that has a limited number of different values. A field in which you intend to store values representing company departments would be an appropriate choice for a coded field. You would soon realize that typing the full name of a department (e.g., *MARKETING*, *TECHNICAL SERVICES*) each time a record is added or updated is time-consuming as well as error-prone. So you would probably choose to enter numeric codes (e.g., 055) for the different departments or appropriate abbreviations (e.g., *MKT*, *TS*). Then, because these codes are often not very meaningful, descriptions of the coded values can be stored for use in reports.

When you use coded fields, not only can you enter data more quickly and accurately, you also reduce the storage space required and improve searching efficiency.

When you are changing a field from un-coded to coded, MABRA asks you to specify the width of the codes' descriptions. In the case of *PAYSTATUS*, there will only be two code descriptions—*PAID* and *UNPAID*. *UNPAID*, the widest of the two descriptions, is 6 elements (characters) wide. Based on this maximum width, MABRA then calculates and displays the maximum number of code descriptions that a code description table for the field could hold. You are then asked if the suggested maximum number of codes is acceptable. If you indicate that it is acceptable by responding *Y*, MABRA goes ahead and changes the field definition to coded. On the other hand, if you planned on having more than this maximum number of codes, you would enter *N*, and MABRA would prompt again for the width of the codes' descriptions. To handle a larger number of codes, you must shorten the width of the descriptions.

In the example above, 1307 well exceeds the number of descriptions that will be required for *PAYSTATUS*.

If, at any time, you want to change the width of the codes' descriptions, you use the *CF* command. For example, suppose you decide to change the code descriptions for *PAYSTATUS* to *WITH PAY* and *WITHOUT PAY*. You would have to change the description width from 6 to 11.

```
07:CF
* CHANGE FIELD
* FIELD NAME: PAYSTATUS
* CHANGE FIELD=PAYSTATUS
* CHANGE NAME,WIDTH,INVERT,CODE,FMT: CODE
* FIELD'S CODE DEFINITION IS 'CODED'.
* UN-CODE FIELD OR CHANGE WIDTH OF CODES' DESCRIPTIONS? (U/W): W
* CURRENT CODE DESCRIPTION WIDTH IS: 6
* NEW WIDTH OF CODES' DESCRIPTIONS: 11
* SUGGESTED MAXIMUM NUMBER OF CODES IS: 720
* IS 720 OK? (Y/N): Y
* WIDTH OF CODES' DESCRIPTIONS CHANGED.
* CHANGE FIELD=PAYSTATUS
* CHANGE NAME,WIDTH,INVERT,CODE,FMT: BACK
07:
```

Defining a coded field actually requires two steps: changing the field's attribute to *CODED* and then adding the necessary codes and descriptions to the field's **code description table**. The second step is described in one of the following sections.

Listing Fields

When you want to display any information regarding the fields in a MABRA system, you use the *LF* (List Fields) command.

The following dialogue with MABRA illustrates the use of *LF*. Fields from system 7 are listed.

```
07:LF
* LIST FIELD DEFINITIONS
* FIELDS: HELP
ENTER ANY OF: EMPLOYEE,STARTDATE,DURATION,PAYSTATUS,Δ,ΔCOUNT, OR ALL
* FIELDS: ALL
* ATTRIBUTES: HELP
ENTER ANY OF: T(TYPE),W(WIDTH),I(INV),D(DESC),B(BYTES),
              F(FMT),E(EXPRESSION), OR ALL
* ATTRIBUTES: ALL
* ALIGN PAPER, TYPE SPACE, CR TO START.
                                                    DATE:  9/08/83

                        07: EMPLOYEE VACATION SYSTEM

CLASS     NAME        TYPE   WIDTH  INV  DESC  BYTES  USED  FMT          EXPR
                                        WID   MAS    INV
-----     ----        ----   -----  ---  ----  -----  ----  ---          ----
REAL      EMPLOYEE    CHAR       4          4.0          4A1
          STARTDATE   DATE       1          4.0          1G□ZZ/ZZ/Z9□
          DURATION    INTE       1          4.0          1I10
          PAYSTATUS   BINA       1     11   0.1          1I1

SESSION   Δ           INTE       1                       I3           (NUM,1)ρRCDS/⍳ρRCDS
PSEUDO    ΔCOUNT      BINA       1                       I1           1

07:STOP
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

When you request a listing of fields, you are first prompted for the names of the fields you wish to include in the listing. You must then enter a list of field names, separated by commas. To find out what fields currently exist in the system, you can enter *HELP*. If you want all fields to be listed, as in the above example, you can enter *ALL* in response to the prompt for *FIELDS*.

The next prompt is for *ATTRIBUTES*. Attributes are ways in which a field is described, and they are defined when you add or change a field. Field name, width, and data type are examples of attributes. In the example above, *HELP* was entered in response to this prompt to obtain a list of the different field attributes. When you enter a list of one or more field attributes, you can type either the attribute's full name (e.g., *TYPE*) or an acceptable abbreviation (e.g., *TYP* or *T*). If you want all field attributes displayed, you enter *ALL*, as in the example above.

When you request a complete listing of field definitions, MABRA prints a table that includes the following field attributes: *CLASS*, *NAME*, *TYPE*, *WIDTH*, *INV*, *DESC WID*, *BYTES USED*, *FMT*, and *EXPR*. Each column contains a different attribute, and each line within the table shows whatever attributes have been defined for a particular field. The different attributes are described below:

CLASS — There are two main field classifications—*REAL* and *PSEUDO*. *REAL* fields are not to be confused with *REAL* data types. Referring to *field*, *REAL* means that the field values are stored in a file. Referring to *data type*, *REAL* means that the field contains real number values (i.e., decimal numbers and very large or very small numbers).

PSEUDO fields contain values that are computed rather than stored. There are two kinds of *PSEUDO* fields—*SESSION* and *RETAINED*. A *SESSION PSEUDO FIELD* disappears when you end your MABRA session, while a *RETAINED PSEUDO FIELD* remains in your system for future use. Every time you enter a system, MABRA automatically adds two session pseudo fields, Δ and Δ*COUNT*. The Δ field contains the MABRA record number which is used when you want to locate individual records quickly. Δ*COUNT* contains the value 1 for every record, and, as the name implies, it is used to count records. Chapter 4 showed how often these fields are used in locating records and producing listings and reports. Pseudo fields are discussed in more detail later in this chapter.

NAME — The name assigned to a field.

TYPE — The field's data type (character, integer, real, binary, or date).

WIDTH — The number of elements in a field.

INV — A field may be inverted for more efficient searching. (Inverted fields are discussed in more detail later in this chapter.)

DESC WID — If a field is coded, it will have a description width. This is the number of elements in each of the fields descriptions. (Coded fields are discussed in more detail later in this chapter.)

BYTES USED — The amount of space used to store a *REAL* field value for one record. This column is blank for all pseudo fields. *MAS* indicates the storage required in MABRA's master file, while *INV* indicates the storage required in MABRA's inverted file. The *INV* file stores only inverted field values.

FMT — The format statement that controls how a field's values will appear when they are printed in listings and reports.

EXPR — The APL expression used to compute the values for a pseudo field.

In the next example, several fields from system 1 are listed.

```
:ENTER;1
* LAST UPDATE:  9/08/83 21:38:06; CPU USED: 8
* NUMBER OF COMPLETED UPDATES: 1
* DATA MODE, SYSTEM IS 01: PERSONNEL
01:LF
* LIST FIELD DEFINITIONS
* FIELDS: FIRSTNAME,IDNUMBER,GROSSPAY
* ATTRIBUTES: ALL
* ALIGN PAPER, TYPE SPACE, CR TO START.
```

```
                                                        DATE:  9/08/83
                          01: PERSONNEL

CLASS      NAME        TYPE   WIDTH  INV  DESC  BYTES  USED  FMT          EXPR
                                          WID   MAS    INV
-----      ----        ----   -----  ---  ----  -----  ----  ---          ----
REAL       FIRSTNAME   CHAR      20        20.0        20A1
           IDNUMBER    CHAR       4         4.0        4A1
           GROSSPAY    INTE       1         4.0        1K-2F10.2

01:STOP
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

The changes made to these fields with the *CF* command are reflected in the above display. *FIRSTNAME* now has a field width of 20 elements, *IDNUMBER* is a non-inverted field, and *GROSSPAY* is now a *REAL* field type with the format statement `1K-2F10.2` governing the output appearance of its values.

Code Description Tables

Because the values stored in a coded field are often not very informative, it is necessary to store a "description" corresponding to each different code or field value for use in reports. These descriptions are stored in a **code description table**.

As soon as you have defined a field as coded, the next step is to add the necessary codes and descriptions to the field's code description table. The *AC* command, illustrated in the following example, is used for this purpose. The table should have a code and a description for each different value that the field may contain.

```
* SYSTEM MODE
:ENTER;7
* DATA MODE, SYSTEM IS 07: EMPLOYEE VACATION SYSTEM
07:AC
* ADD CODES/DESCS.
* FOR WHICH FIELD: PAYSTATUS
* CURRENT NUMBER OF CODES IS 0.
* MAXIMUM NUMBER OF CODES IS 720.
* CODE: 0
* DESC: WITHOUT PAY
* CODE: 1
* DESC: WITH PAY
* CODE: STOP
* UPDATED.
07:
```

After entering *AC*, you are prompted first for the name of the coded field whose description table you wish to update. This example added two code descriptions to the table for *PAYSTATUS*.

Then MABRA tells you the number of codes that are currently stored in that field's description table, as well as the maximum number of codes that the table can hold. After this information is displayed, you are prompted for a code and then for its description. Code values must, of course, be the same data type and the same width as the field. In this case, *PAYSTATUS* is a one-element binary field. A

code for a character field can consist of any valid symbol that constitutes character data—any letter of the English alphabet, any numeral (0 through 9), or special symbol (e.g., /, -).

If you code a field that has more than one element, all codes must have the same number of elements. Each unique combination of elements then makes a single code. For example, if you code a two-element character field whose values represent countries of the world, each of the combinations *CA*, *UK*, *US*, and *FR* would be a distinct code requiring a distinct code description.

The width of each code description must not exceed the width specified as the maximum description width when the field was coded.

If you want to change the description for a code that has just been entered or one that is already in the table, you simply enter that code again and give a new description. The new description will then replace the old one.

MABRA continues to prompt for codes and descriptions until you type *STOP* in response to the request for a *CODE*. A message is then displayed indicating that the description table has been updated.

To list a field's code description table, you use the *LC* command.

```
07:LC
* LIST CODES/DESCS
* FOR WHICH FIELD: PAYSTATUS
* SORT BY CODE OR DESCRIPTION (C/D): C
* ALIGN PAPER, TYPE SPACE, CR TO START.

* CURRENT NUMBER OF CODES IS 2.
* SUGGESTED MAXIMUM NUMBER OF CODES IS 720.

0  WITHOUT PAY
1  WITH PAY

* FOR WHICH FIELD: STOP
07:STOP
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

When you enter *LC*, MABRA asks you to give the name of the field whose description table you wish to see. Then you must indicate how you want the listing sorted—by code or by description.

The listing of a code description table always begins with the number of codes currently in the table and the suggested maximum number of codes that the table can hold. Then the codes and descriptions are printed, sorted in the manner you requested.

After the listing is printed (and you type SPACE followed by RETURN), MABRA prompts for another field name. If you have finished displaying code description tables, you would enter *STOP* in response to this prompt.

In the next example, we return to system 1 and delete a code from the description table for *EEO*. The *DC* command performs this function.

```
:ENTER;1
* LAST UPDATE:  9/08/83 21:38:06; CPU USED: 8
```

```
* NUMBER OF COMPLETED UPDATES: 1
* DATA MODE, SYSTEM IS 01: PERSONNEL
01:DC
* DELETE CODES/DESCS
* FOR WHICH FIELD: EEO
* CODE: M
* CODE: STOP
* UPDATED.
01:
```

After you have entered *DC*, you must specify the field whose description table needs to be modified. Then you are prompted one at a time for the codes to be deleted. When you delete a code from a field's code description table, you also remove the description for that code.

When you have entered all the codes you wish to delete, you enter *STOP* and the description table is updated.

Next, you decide to display the list of codes and related descriptions for *EEO*.

```
01:LC
* LIST CODES/DESCS
* FOR WHICH FIELD: EEO
* SORT BY CODE OR DESCRIPTION (C/D): C
* ALIGN PAPER, TYPE SPACE, CR TO START.

* CURRENT NUMBER OF CODES IS 5.
* SUGGESTED MAXIMUM NUMBER OF CODES IS 500.

A  AMERICAN INDIAN
B  BLACK
H  HISPANIC
O  ORIENTAL
W  WHITE

* FOR WHICH FIELD: STOP
01:
```

Defining Access to a MABRA System

When you create a MABRA system on your account number, you are the *CREATOR*, and only your account number can use the system. Only your account can add data, modify it, and display it. If you want to give other account numbers access to your system, you have to use the commands described in this section. There are commands that allow you to set access (*SA*) for other account numbers, list access (*LA*), change access (*CA*), and remove access (*RA*).

In MABRA, there are three different types of access that must be defined:

Command access	the commands that another account can use when working with your system.
Record access	the records in the system that can be worked with.
Field access	the fields within each record that can be worked with.

In the next example, the *SA* command is used to give account 1729760 access to system 1. This account number belongs to the new assistant personnel director, who needs extensive access to the system.

```
01:SA
* SET ACCESS
* USER(S): 1729760
* USER(S) ARE: 1729760. OK? (Y/N): Y
* COMMAND ACCESS
* COMMANDS: ALL
* RECORD ACCESS
* CONSTRAINTS: ALL
   196 FOUND.
* IS THIS CONSTRAINT OK? (Y/N): Y
* FIELD ACCESS
* AR FIELD ACCESS
* FIELDS: ALL
* MORE? (Y/N): N
* CR AND CD FIELD ACCESS
* FIELDS: ALL
* MORE? (Y/N): N
* LR FIELD ACCESS
* FIELDS: ALL
* MORE? (Y/N): N
* SET ACCESS? (Y/N): Y
* ACCESS SET
* USER(S):
```

When you use the *SA* command, you first have to specify the account number (*USER*) that you are setting the access for. If you wanted to give the same access to more than one account number, you would enter a list of account numbers, each account number separated from the next by a blank space.

If you enter an account number that already has some level of access, the new access that you set will *replace*, rather than *add to*, the existing access.

Once you have confirmed the account number(s), you proceed to define the command access. When MABRA prompts for *COMMANDS*, you enter one or more of the following *DATA MODE* commands, separated by commas: *AN*, *AP*, *AR*, *BP*, *CD*, *CR*, *DR*, *GR*, *LC*, *LF*, *LR*, *LS*, *LU*, *NS*, *PF*, *SF*, *UP*, or *WU*. These are the only commands that MABRA will permit you to share with another account. Only the *CREATOR* can use the following commands when working with a MABRA system: *AC*, *AF*, *CA*, *CF*, *CS*, *DC*, *DF*, *FO*, *LA*, *RA*, *RB*, *RS*, *SA*, and *ST*. If you wish to give an account access to all of the sharable commands, you can enter *ALL*, as in the example above.

Next, you are asked to define record access. This involves entering a constraint that specifies which records within the system are to be accessible to the account. The account will then be able to select all existing and future records that meet the constraint. Access to *ALL* records was given to account 1729760.

After showing how many records currently meet the constraint, MABRA prompts for your approval. If you indicate that the constraint is not correct, you are prompted for a new constraint.

The last step in the process of setting access for an account is defining the field access. There are three types of field access that may need to be specified:

AR FIELD ACCESS	fields that can be added using *AR*.
CR AND CD FIELD ACCESS	fields that can be changed using *CR* or *CD*.
LR FIELD ACCESS	fields that can be listed using *AN*, *AP*, *BP*, *CD*, *CR*, *DR*, *GR*, *LR*, *LS*, or *PF*.

If you do not give an account access to at least one of the commands related to each type of field access, you are not prompted to define that type of access. Thus, if you did not give an account access to the *AR* command, you would not be asked to define *AR FIELD ACCESS*.

Access to *ALL* fields is given to account 1729760 for all three types of field access.

Once field access has been defined, MABRA asks if the access can now be set. If you are satisfied that all access specifications have been entered correctly, you would tell MABRA to proceed with setting the access. You are then prompted again for *USER(S)*. At this point, you could set access for another account or group of accounts, or you could enter *STOP* to exit from the *SA* command.

If you tell MABRA not to set the access, all access specifications just entered are ignored, and you are prompted for *USER(S)*. You then have the opportunity to again define the access specifications for an account or group of accounts.

In the next example, access is set for an account that belongs to a data entry clerk. This account is given limited access to system 1.

```
* USER(S): 5472893
* USER(S) ARE: 5472893. OK? (Y/N): Y
* COMMAND ACCESS
* COMMANDS: AR
* RECORD ACCESS
* CONSTRAINTS: ALL
   196 FOUND.
* IS THIS CONSTRAINT OK? (Y/N): Y
* FIELD ACCESS
* AR FIELD ACCESS
* FIELDS: LASTNAME,FIRSTNAME,INITIAL,BIRTHDATE
* MORE? (Y/N): Y
* FIELDS: SEX,IDNUMBER
* MORE? (Y/N): N
* SET ACCESS? (Y/N): Y
* ACCESS SET
* USER(S):
```

According to the access set above, account 5472893 will only be able to add certain field values to the system. It will not be able to list or change any fields from any records.

Because you only gave the account access to the *AR* command, MABRA only prompted you for *AR FIELD ACCESS* (and not the other two types of field access). When you are entering field names, the names must be separated by commas. Because you may have many field names to enter for a particular type of field access, MABRA continues to prompt for more fields until you indicate that you have no more to enter.

This account will only be permitted to add values for six of the system's fields. The more confidential fields (*EEO*, *SALARY*, *EXEMPT*, and *GROSSPAY*) are not included in the list of fields entered for *AR FIELD ACCESS*. Thus, when this account adds records to system 1, MABRA will only prompt for values

for *LASTNAME*, *FIRSTNAME*, *INITIAL*, *BIRTHDATE*, *SEX*, and *IDNUMBER*. All other fields will be given fill values for any records added by this account.

In the last example of the *SA* command, access is set for an account that belongs to a local personnel manager in the Canadian branch office. This user needs to have access to all permissible *DATA MODE* commands for all fields, but only for the records of employees in that branch.

```
* USER(S): 3674762
* USER(S) ARE: 3674762. OK? (Y/N): Y
* COMMAND ACCESS
* COMMANDS: ALL
* RECORD ACCESS
* CONSTRAINTS: IDNUMBER=2
    28 FOUND.
* IS THIS CONSTRAINT OK? (Y/N): Y
* FIELD ACCESS
* AR FIELD ACCESS
* FIELDS: ALL
* MORE? (Y/N): N
* CR AND CD FIELD ACCESS
* FIELDS: ALL
* MORE? (Y/N): N
* LR FIELD ACCESS
* FIELDS: ALL
* MORE? (Y/N): N
* SET ACCESS? (Y/N): Y
* ACCESS SET
* USER(S): STOP
01:
```

The first element of the *IDNUMBER* field identifies the branch that an employee works in. So, when you entered the constraint *IDNUMBER*=2 to define the record access for account 3674762, you were limiting the access to current and future records of all employees in the Canadian branch office.

When other user accounts have the ability to initiate update tasks for a MABRA system that you own, your sign-on password must be stored in a locked function in a file on your account. This is necessary because MABRA uses this function to start up an update task on *your* account, whenever an update task is initiated by an account to which you have given *UP* access.

The *NEWPASS* command is used to store your password, and it can only be entered in *SYSTEM MODE*. The following example illustrates the use of this command.

```
01:STOP
* MABRA VERSION 3.1
* SYSTEM MODE
:NEWPASS
* ENTER YOUR SIGN-ON PASSWORD: [redacted]
* PASSWORD UPDATED.
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

You would type your password after MABRA has printed the security blot. From then on, any time

you change your sign-on password, you would also have to use *NEWPASS* to store the new password, and thus enable the other users to continue initiating update tasks.

To list the access that has been defined for a system, you use the *LA* command. You have a choice of listing access by user account number or by access level. Examples of each type of listing follow.

```
* SYSTEM MODE
:ENTER;1
* LAST UPDATE:  9/08/83 21:38:06; CPU USED: 8
* NUMBER OF COMPLETED UPDATES: 1
* DATA MODE, SYSTEM IS 01: PERSONNEL
01:LA
* LIST ACCESS
* LIST BY USER OR BY ACCESS LEVEL? (U/L): U
* ENTER USER NUMBER(S): ALL
* ALIGN PAPER, TYPE SPACE, CR TO START.

* USER: 1234567

  COMMANDS: LR,LF,CR,CF,AR,AF,DR,DF,AN,RS,RB,NS,SF,PF,SA,LA,RA,UP,AC,LC,DC,
            ST,CS,LU,WU,BP,FO,CA,CD

  RECORD CONSTRAINT: ALL

  FIELD              ADD CHA LIS
  LASTNAME             X   X   X
  FIRSTNAME            X   X   X
  INITIAL              X   X   X
  BIRTHDATE            X   X   X
  SEX                  X   X   X
  SALARY               X   X   X
  EEO                  X   X   X
  IDNUMBER             X   X   X
  EXEMPT               X   X   X
  ADDRESS2             X   X   X
  CITY                 X   X   X
  ZIP                  X   X   X
  ADDRESS1             X   X   X
  HIREDATE             X   X   X
  DEPT                 X   X   X
  GROSSPAY             X   X   X
-------------------------------

* USER: 1729760

  COMMANDS: LR,LF,CR,AR,DR,AN,NS,SF,PF,UP,LC,LU,WU,BP,CD,AP,GR,LS

  RECORD CONSTRAINT: ALL

  FIELD              ADD CHA LIS
  LASTNAME             X   X   X
  FIRSTNAME            X   X   X
  INITIAL              X   X   X
  BIRTHDATE            X   X   X
```

```
  SEX                  X   X   X
  SALARY               X   X   X
  EEO                  X   X   X
  IDNUMBER             X   X   X
  EXEMPT               X   X   X
  ADDRESS2             X   X   X
  CITY                 X   X   X
  ZIP                  X   X   X
  ADDRESS1             X   X   X
  HIREDATE             X   X   X
  DEPT                 X   X   X
  GROSSPAY             X   X   X
--------------------------------

* USER: 5472893

  COMMANDS: AR

  RECORD CONSTRAINT: ALL

  FIELD               ADD CHA LIS
  LASTNAME             X
  FIRSTNAME            X
  INITIAL              X
  BIRTHDATE            X
  SEX                  X
  IDNUMBER             X
--------------------------------

* USER: 3674762

  COMMANDS: LR,LF,CR,AR,DR,AN,NS,SF,PF,UP,LC,LU,WU,BP,CD,AP,GR,LS

  RECORD CONSTRAINT: IDNUMBER=2

  FIELD               ADD CHA LIS
  LASTNAME             X   X   X
  FIRSTNAME            X   X   X
  INITIAL              X   X   X
  BIRTHDATE            X   X   X
  SEX                  X   X   X
  SALARY               X   X   X
  EEO                  X   X   X
  IDNUMBER             X   X   X
  EXEMPT               X   X   X
  ADDRESS2             X   X   X
  CITY                 X   X   X
  ZIP                  X   X   X
  ADDRESS1             X   X   X
  HIREDATE             X   X   X
  DEPT                 X   X   X
  GROSSPAY             X   X   X
--------------------------------
```

```
* ENTER USER NUMBER(S): STOP
* LIST BY USER OR BY ACCESS LEVEL? (U/L):
```

An access listing by user includes the account number, the commands that can be used by the account, the record constraint that defines which records are accessible, and the fields that the account can add, change, and list.

When you are prompted for *USER NUMBER(S)*, you can enter one account number, a list of account numbers separated by blanks, or *ALL* to obtain a listing that includes the access for all accounts with whom you are sharing your system.

The first account listed in the example above is the *CREATOR*'s account which of course has complete access to the system. The next account belongs to the assistant personnel director who has extensive access to the system, and this is followed by the account of the data entry clerk who is limited to one command and six fields. Finally, the access assigned to the Canadian personnel manager's account is listed. He has access to all sharable commands, but can only work with records in which 2 is the first element of the *IDNUMBER* field.

The next example shows how a listing by access level is produced.

```
* LIST BY USER OR BY ACCESS LEVEL? (U/L): L
* LIST BY RECORD, FIELD, OR COMMAND ACCESS? (R/F/C): R
* ALIGN PAPER, TYPE SPACE, CR TO START.

USER        RECORD ACCESS

   1234567 ALL
   1729760 ALL
   3674762 IDNUMBER=2
   5472893 ALL

* LIST BY RECORD, FIELD, OR COMMAND ACCESS? (R/F/C): F
* ALIGN PAPER, TYPE SPACE, CR TO START.

                1234567 1729760 3674762 5472893
LASTNAME            ACL     ACL     ACL       A
FIRSTNAME           ACL     ACL     ACL       A
INITIAL             ACL     ACL     ACL       A
BIRTHDATE           ACL     ACL     ACL       A
SEX                 ACL     ACL     ACL       A
SALARY              ACL     ACL     ACL
EEO                 ACL     ACL     ACL
IDNUMBER            ACL     ACL     ACL       A
EXEMPT              ACL     ACL     ACL
ADDRESS2            ACL     ACL     ACL
CITY                ACL     ACL     ACL
ZIP                 ACL     ACL     ACL
ADDRESS1            ACL     ACL     ACL
HIREDATE            ACL     ACL     ACL
DEPT                ACL     ACL     ACL
GROSSPAY            ACL     ACL     ACL

* LIST BY RECORD, FIELD, OR COMMAND ACCESS? (R/F/C): C
* ALIGN PAPER, TYPE SPACE, CR TO START.
```

```
    USER COMMANDS

  1234567 LR,LF,CR,CF,AR,AF,DR,DF,AN,RS,RB,NS,SF,PF,SA,LA,RA,UP,AC,LC,DC,ST,
          CS,LU,WU,BP,FO,CA,CD
  1729760 LR,LF,CR,AR,DR,AN,NS,SF,PF,UP,LC,LU,WU,BP,CD,AP,GR,LS
  3674762 LR,LF,CR,AR,DR,AN,NS,SF,PF,UP,LC,LU,WU,BP,CD,AP,GR,LS
  5472893 AR
* LIST BY RECORD, FIELD, OR COMMAND ACCESS? (R/F/C): BACK
01:
```

When you request a listing by access level, you are asked to specify whether you want to see record, field, or command access. In the example above, record access is displayed first, followed by field access and command access.

The listing of record access includes the record constraint for each account that defines which records are accessible. When field access is listed, you can see what fields an account can add, list, or change. The listing for command access shows what commands each account can use.

To change the access for one or more accounts, you use the *CA* (Change Access) command. In the following example, *CA* is used to change the access for account 5472893 which belongs to the data entry clerk. You want this account to have access to the *CD* as well as the *AR* command.

```
01:CA
* CHANGE ACCESS
* USER(S): 5472893
* CHANGE: COMMAND, RECORD, OR FIELD ACCESS? (C/R/F): C
* CHANGE COMMAND ACCESS
* COMMANDS: AR,CD
* CHANGE: COMMAND, RECORD, OR FIELD ACCESS? (C/R/F): F
* FIELD ACCESS
* CHANGE: ADD, CHANGE, OR LIST FIELD ACCESS? (A/C/L): C
* CHANGE CR FIELD ACCESS
* FIELDS: LASTNAME,FIRSTNAME,INITIAL,BIRTHDATE
* MORE FIELDS? (Y/N): Y
* FIELDS: SEX,IDNUMBER
* MORE FIELDS? (Y/N): N
* CHANGE: ADD, CHANGE, OR LIST FIELD ACCESS? (A/C/L): STOP
* CHANGE: COMMAND, RECORD, OR FIELD ACCESS? (C/R/F): STOP
* OK TO CHANGE ACCESS? (Y/N): Y
* ACCESS CHANGED.
* USER(S): STOP
01:
```

When you use the *CA* command, you are first prompted for the account number(s) for which the access needs to be changed. Next, you must specify which types of access you want to modify. MABRA then prompts for the details needed to change that type of access.

Note that when you change any type of access, the new access that you define supersedes (rather than adds to) the existing access for that type. That is why, in the example above, both *AR* and *CD* were entered when MABRA prompted for *COMMANDS*. If only *CD* had been entered, account 5472893 would no longer have *AR* access.

Because you added *CD* access to account 5472893, the *CR AND CD FIELD ACCESS* needed to be changed as well. The other two types of field access did not need to be altered in any way.

The *CA* command is appropriate for use when you only want to change a *portion* of user access that has already been set with *SA*. If you wish to make extensive changes to the access defined for one or more accounts, it may be easier to use the *SA* command and set new access specifications.

To make the same access change(s) for all users who currently have access to your system, you can enter *ALL* in response to the prompt for *USER(S)*. The access for the *CREATOR* of the system cannot be changed.

When you have entered all changes for the specified user account(s), you type *STOP* in response to the *CHANGE: COMMAND, RECORD, OR FIELD ACCESS* prompt. MABRA then asks for confirmation and, depending on your response, either proceeds to make the changes or returns to the *USER(S)* prompt without making changes. In either case, you then have the opportunity to change the access for another user account or group of accounts.

You terminate the *CA* command by entering *STOP* in response to the *USER(S)* prompt.

To remove access for one or more accounts, you use the *RA* (Remove Access) command. In the following example, access is revoked for account 3674762.

```
01:RA
* REMOVE ACCESS
* USER(S): 3674762
* USER(S): 3674762. OK? (Y/N): Y
* ACCESS REMOVED.
01:
```

When you use the *RA* command, you are asked to specify the account number(s) that you want to remove the access for. Account numbers must be separated by a blank space if you enter more than one.

MABRA confirms when the access has been removed.

Users whose access is revoked while they are using the system will get a *FILE ACCESS ERROR* the next time they enter a command.

Locking a MABRA System

When other users have been given access to your MABRA system, you may want to add an extra level of security by putting a lock on the system. *LOCK*, another of MABRA's *SYSTEM MODE* commands, is used for this purpose. In the following example, this command is used to put a lock on system 1.

```
* SYSTEM MODE
:LOCK
* LOCK SYSTEM.
* SYSTEM NUMBER: 1
* SYSTEM IS NOT LOCKED.
* LOCK? (Y/N): Y
* SYSTEM LOCKED.
* SYSTEM NUMBER: STOP
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

Resizing a MABRA System

When you first create a MABRA system, you specify a maximum number of records that you expect the system to hold. If, as you work with the system, you find it necessary to change this maximum number, you can use the *RS* (Resize System) command.

The following example shows how *RS* is used to increase the capacity of system 1 so the records of 1000 employees can be accommodated.

```
01:RS
* RESIZE SYSTEM
* ENTER MAXIMUM NUMBER OF RECORDS EXPECTED: 1000
* RESIZE DONE
01:
```

The *RS* command also determines how many file components should be set aside for inverted fields. So, if any fields are inverted when you use *RS*, MABRA reorganizes the inverted file to provide more or fewer components for those fields.

To be economical, it is wise to avoid frequent use of *RS* by setting the system size liberally. Setting a system size that is larger than required does not affect the efficiency of computer processing or storage.

Reblocking the Master File of a MABRA System

The master file stores all the records in a MABRA system. When you first add fields to a system, MABRA structures the records in the master file into blocks of equal size. The **block size** is the amount of data MABRA stores in each block. The *RB* (Reblock Master File) command lets you change the block size.

The other function of the *RB* command is to remove deleted records from the master file. When you delete records using the *DR* command, MABRA doesn't remove these records from storage, but simply ignores them. By reblocking, however, you can purge or remove all deleted records from the master file.

Because purging records consumes computer time, it is not efficient to use *RB* for this purpose every time you delete a few records. However, if your system contains a large percentage of deleted records, purging them via *RB* does make searching and retrieval more efficient. It also decreases your computer storage requirements.

The following example shows how *RB* is used to reblock system 1.

```
01:RB
* REBLOCK MASTER FILE
* NUMBER OF RECORDS (INCLUDING DELETED) =201
* NUMBER OF DELETED RECORDS =5
* PURGE DELETED RECORDS? (Y/N): Y
* NOTE: PURGING DELETED RECORDS CHANGES ABSOLUTE RECORD NUMBERS (DELTA NUMBERS)
* CALCULATED BLOCKING FACTOR=211. DO YOU WISH TO INPUT YOUR OWN? (Y/N): N
```

```
* REBLOCKING DONE
01:STOP
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

When you invoke the *RB* command, you are first told how many records are currently stored in the master file (including deleted records). MABRA then asks if you wish to purge the deleted records. If you respond that you do want these records purged, you are warned that the Δ record numbers will change.

Δ record numbers, as we have seen, can be used in record constraints as a quick and efficient way to locate individual records. They actually represent the location of records within a MABRA system. Purging deleted records changes the location of records within the master file, so record numbers also change. If you use record numbers in search constraints, you would need to generate new listings of Δ record numbers after purging deleted records.

If you must have fixed record numbers for your application, there is a utility that inhibits the purging of deleted records during *RB*. It is described in the MABRA *Programmer's Guide*.

Next, the calculated blocking factor is displayed, and you are asked if you wish to override this blocking factor and input your own. You may, for example, need to reblock to a lower blocking factor as recommended by the *AF* command when fields are added.

MABRA confirms when the reblocking has been completed. For further information on reblocking, refer to the MABRA *Programmer's Guide*.

Renaming a MABRA System

If, at any time, you decide that the name given to a MABRA system is inappropriate or needs to be changed for some reason, you can use the *RENAME* command. This is one of MABRA's commands that can only be used in *SYSTEM MODE*.

In the following example, *RENAME* is used to change the name of system 7 from *EMPLOYEE VACATION SYSTEM* to *VACATION FILE*.

```
* SYSTEM MODE
:RENAME
* RENAME SYSTEM
* SYSTEM NUMBER TO BE RENAMED: 7
* NEW NAME: VACATION FILE
* YOU WISH TO CHANGE THE NAME OF SYSTEM 07 FROM 'EMPLOYEE VACATION SYSTEM'
  TO 'VACATION FILE'.
* OK TO RENAME? (Y/N): Y
* DONE.
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

When you use *RENAME*, you must enter the number of the system you wish to rename and the new name. The system number is not affected by the *RENAME* command.

Once you have confirmed that the new name has been entered correctly, MABRA makes the change. From then on, whenever reports are produced from this system, the new name is used.

Displaying the Status of a MABRA System

The *ST* command is used to generate a report that gives the vital statistics of a MABRA system. The following example shows what details the report includes.

```
* SYSTEM MODE
:ENTER;1
* LAST UPDATE:  9/08/83 21:38:06; CPU USED: 8
* NUMBER OF COMPLETED UPDATES: 1
* DATA MODE, SYSTEM IS 01: PERSONNEL
01:ST
* SYSTEM STATUS
            01: PERSONNEL     10/08/83
 TOTAL NUMBER OF RECORDS: 196
 NUMBER OF DELETED RECORDS: 0
 MAXIMUM NUMBER OF RECORDS EXPECTED: 1000
 RECORD SIZE: 169.25 (BYTES)
 MASTER FILE'S BLOCKING FACTOR: 211
 NUMBER OF REAL FIELDS: 16
 NUMBER OF RETAINED-PSEUDO FIELDS: 0
 NUMBER OF SESSION-PSEUDO FIELDS: 2
 NUMBER OF INVERTED FIELDS: 4
 NUMBER OF CODE/DESC. TABLES: 3
 FILE STATUS
      MAS01:     41,028 (BYTES)
      INV01:     88,368 (BYTES)
      DIR01:     37,872 (BYTES)
      UPD01:      6,312 (BYTES)
      ACC01:     12,624 (BYTES)
      TOTAL:    186,204 (BYTES)

 NUMBER OF PENDING UPDATES: 0
 FIRST PENDING UPDATE REQUEST NUMBER: 9
 NUMBER OF COMPLETED UPDATES: 1
 TOTAL NUMBER OF UPDATES: 1
 THE FOLLOWING USERS HAVE ACCESS TO THE SYSTEM: 1234567 1729760 5472893
 ACTIVE USERS: 1234567
01:STOP
* MABRA VERSION 3.1
* SYSTEM MODE
:
```

The report begins with the system's number and name, followed by the current date. Next is the total number of records (including deleted records that have not been purged) stored in the system, the number of records that have been deleted but not yet purged (this number will be 0 immediately after deleted records are purged with *RB*), and then the maximum number of records that the system can store (this number is set by the *CREATOR* of the system when the system is first created or when it is resized).

The next statistics pertain to record size and blocking factor. Each field of a record takes a certain

amount of storage space depending on the type of data stored and the different field widths. When the storage space (which is measured in bytes) for all fields is added, the result is the record size. The blocking factor is the number of records stored in one block of the master file. In this example, 211 records make up one block.

A number of details concerning the system's fields follow next. These include the number of real fields (fields that contain data that must be stored in the master file), retained pseudo fields, and inverted fields. A code description table exists for each field that you have defined as coded, and the total number of tables is included in the status report.

The next part of the report shows how much storage space your MABRA system uses. Every system has the following five files (the system number is used as the last two characters of each file name):

*MAS*01 is the master file where all records are stored.

*INV*01 is the inverted file where all inverted field values are stored.

*DIR*01 is the directory file that stores miscellaneous data of interest primarily to programmers.

*UPD*01 is the update file where update requests are stored. The *RC* command removes completed update requests from this file.

*ACC*01 is the access file which stores all information pertaining to command, record, and field access granted to other user accounts.

The report shows the amount of storage (in bytes) currently used by each of the above files.

Update statistics make up the next part of the report. The number of pending update requests is given, as well as the number assigned to the first pending update request. The number of completed update requests and the total number of update requests (completed and pending) stored in the update file are included as well.

Finally, the report lists the account numbers of users who have access to the system and the account numbers of users who are currently accessing the system.

Erasing a MABRA System

You can erase a MABRA system that is no longer needed with the *ERASE* command. *ERASE* is another one of MABRA's *SYSTEM MODE* commands.

In the following example, system 2 is erased.

```
* SYSTEM MODE
:ERASE
* ERASE SYSTEM
* ENTER SYSTEM NUMBER OF SYSTEM TO BE ERASED: 2
* THE SYSTEM YOU WISH TO ERASE IS:
* 02: NORTH AMERICAN PERSONNEL
* ERASE? (Y/N): Y
* ARE YOU SURE? (Y/N): Y
* SYSTEM ERASED.
* MABRA VERSION 3.1
* SYSTEM MODE
:STOP
      )OFF
```

INDEX

A

abbreviated data in fields, see *coded fields*
aborting commands, 15
AC command, 95
access file, 109
access to a MABRA system, changing, 104
  defining, 97
  listing, 101
  removing, 105
  setting, 98
accessing, MABRA systems belonging to other users, 16
active users of a MABRA system, 109
adding, codes to a code description table, 95
  field orders, 68
  fields, 81
  form specifications, 54
  pseudo fields, 87
  records, 65
address labels, see *mailing labels*
AF command, 81
aligning paper, for output, 17 39
ALL, use as a constraint, 22
α *alpha*, use with date fields, 21
alternate column headings, 20 35 37
alternate field orders, 68
alternate output format, for fields, 83
AN command, 25
analyzing, data, 25
∧ *and*, use in constraints, 19
answer-ahead definitions, see *stored answer-aheads*
answering ahead, 48 67
AP command, 36 81 87
APL immediate execution mode, returning to, 15
appending records, 79
AR command, 65
ascending sort, 21
← *assignment arrow*, use in changing records, 72
⋆ *asterisk*, see *pagebreak star*
attributes of a field, 93
averaging, data with *AN*, 25 29 31

B

B-tasks, 24 76
BACK, 15
batch printing records, 24
batch task, to print reports, 39
BATCHPRINT, *GR* special option, 39
BINA, data type, 82
binary data, 82 83
binary fields, 82 87
blank columns on a form, trimming, 55 61
blank rows on a form, removing, 55 61
block size, 106
blocking factor, 107 109
BP command, 24 53
[ ] *brackets*, use in constraints, 24
→ *branch to*, 67 68 73
BYTES USED, field attribute, 94

C

CA command, 104
calculation, to determine pseudo field values, 88 89
CD command, 71 72
CF command, 89
changing, access to a MABRA system, 104
  calculation for pseudo fields, 89
  coding status of a field, 91
  column headings, 20 35 37
  CREATOR setting, 17
  data type, 82
  date format, 16
  description for a code, 96
  field attributes, 89
  field class, 91
  field name, 89
  field orders, 68
  field values, 71
  field width, 90
  fields, 89
  form specifications, 62
  format statements, 89 90
  inversion status of a field, 91
  name of a MABRA system, 107
  records, 71
  size of a MABRA system, 106
  stored answer-aheads, 48
  width of code descriptions, 92
CHAR, data type, 82
character data, 82 83
character fields, 82
  format statements for, 84
  use in constraints, 23
checking, update requests, see *listing, update requests*
CLASS, field attribute, 94

classifying, MABRA systems, 16
code description tables, 92 95
code descriptions, 91 95
  changing, 96
  maximum number of, 92 95
  printing in reports, 20 37
  width of, 92 96
coded fields, 91 94 95
: *colon*, use as a prompt, 14 28 71
  use in setting up stored answer-aheads, 46
  use in specifying column headings, 20 35 37
column headings, for reports, 20 35 37
columns, of a crosstabulation analysis, 28 32
  of a form, 54
command access, 97 98
command prompt, in data mode, 14
  in system mode, 14
completed update requests, 74
  removing, 77
computing, data from other field values, see *pseudo fields*
constraints, 19 21
⊃ *contains*, use in constraints, 19 20
copying, fields from another MABRA system, see *linking*
  a MABRA system, 78
  records from a MABRA system, 78
correcting errors, 15
counting records, 13
CR command, 71
CREATE command, 80
creating, a MABRA system, 80
CREATOR, 16
crosstabulation analyses, 25
CS command, 78
custom programming, 5

D

data, computed from other field values, see *pseudo fields*
data mode, 14 18
  command prompt, 14
data type, 82 94
  changing, 82
DATE, data type, 82
date fields, 21 86
  use in constraints, 23
date formats, 16 66
DC command, 96
decorators, for format statements, 84
decreasing, field width, 90
default output format, 83
default report title, 35
defining, access to MABRA systems, 97
  forms, 54
  range points for data, 27
  stored answer-aheads, 46
deleting, codes from a code description table, 96
  completed update requests, 77
  field orders, 68
  fields, 88
  form specifications, 64
  records, 70
  stored answer-aheads, 48
delivery instructions, 24 75
Δ *delta*, session pseudo field containing record numbers, 22 23 94 107
Δ *delta underscore*, use as a prefix, 50
Δ*COUNT*, session pseudo field, 13 23 34 94
DEPTH, *GR* special option, 39
derivatives, of fields, 21
DESC WID, field attribute, 94
descending sort, 21
descriptions, for coded fields, 91 95
DF command, 88
directory file, 109
displaying, see also *listing*
  blocking factor, 109
  maximum number of records for a system, 108
  number of deleted records, 108
  number of records in a MABRA system, 108
  record size, 108
  status of a MABRA system, 108
  storage space used by a MABRA system, 109
  update statistics, 109
DR command, 70

E

86 *MABRA*, workspace, 14
elements in a field, 82 94
ENTER command, 18
entering, data, 65
  data mode, 18
  date field values, 66
  invalid data values, 66
  MABRA systems, 18
  more than one response on a line, 67
  numeric field values, 66
= *equal to*, use in constraints, 19
ERASE command, 109
erasing, a MABRA system, 109
errors, correcting, 15
EXIT, 15

exiting, 86 *MABRA*, 15
a MABRA system, 18
EXPR, field attribute, 94
expression, to determine pseudo field values, 88

F

field, definition of, 3
field access, 97 98
for *AR*, 99
for *CR* and *CD*, 99
for *LR*, 99
field attributes, 93
field averaging, with *AN*, 25 29 31
field class, changing, 91
field code status, 91 94 95
field derivatives, 21
field input validation, 66
field inversion status, 91
field names, 81
changing, 89
field ordering, for *AR*, 68
field output format, 83
field totalling, with *AN*, 25 29 31
with *GR*, 34 39
with *LS*, 43 46
field width, 83
changing, 90
field variations, 21
fields, changing, 89
coded, 91 94 95
inverted, 91
maximum number of, 3
FILE ACCESS ERROR, 105
file space, economizing, 77 82 83 87
FILESORT utility, 39
filler data, for empty fields, 67
FMT, field attribute, 94
FO command, 68
form depth, 39 54 58
form dimensions, 54 58
form letters, 53 58
form width, 54 58
format statements, 83 94
changing, 89 90
forms, defining, 54
inserting record data on, 53
listing, 56
frequency analysis, 25 30 32 37
frequency counts, 25 30 32 37

G

"go to" feature, 67 68 73
GR command, 33
granting, access, see *defining, access*
> *greater than*, use in constraints, 19
≥ *greater than or equal to*, use in constraints, 19
grouping, records in a report, 36 38 39

H

HELP, 14
highspeed print specifications, 24
HSPRINT utility, 25 39 53

I

identification numbers, for field orders, 69
for forms, 54
incorrect responses, 15
increasing, field width, 90
inhibiting, purging of deleted records, 107
initiating, MABRA, 14
an update task, 76
inputting, see *entering*
INTE, data type, 82
integer data, 82 83
INV, field attribute, 94
invalid responses, 15
inverted fields, 91 106
inverted file, 109
item numbers, for forms, 56

J

"jump to" feature, see *"go to" feature*

K

key fields, for linking data, 43

L

LA command, 101
LB3U, print specification, 54
LC command, 96
< *less than*, use in constraints, 19
≤ *less than or equal to*, use in constraints, 19
LF command, 93
linking, character fields, 46

fields from another MABRA system, 40 53
LIST command, 17
listing, access to a MABRA system, 101
code description tables, 96
field orders, 69
fields, 93
forms, 56 61
MABRA systems, 17
records, 19
stored answer-aheads, 47
update requests, 73
locating, records, see *constraints*
LOCK command, 105
locking, a MABRA system, 105
LR command, 19
LS command, 40 53
LU command, 73

M

MABQ, mailbox group, 5
MABRA utilities, 4
MABRA workspace, 14
MABRAUTIL, 4
mailing labels, 53
major sort fields, 21 34
master file, 106 109
reblocking, 106
maximum number of records, in a MABRA system, 78 80 106
merging, MABRA systems, see *copying, a M system*
\- *minus sign*, use in sorting records, 21
multi-line column headings, 20 35 37

N

N-tasks, 24 76
NAME, field attribute, 94
naming, fields, 81
MABRA systems, 80
new snapshot, 77
NEWPASS command, 100
NOERASE, print specifications, 58
non-procedural version of MABRA, 4
NONE, 71
≠ *not equal to*, use in constraints, 19
NOUSDATES, state setting, 16
NS command, 77
number of records, in a MABRA system, 108
numbers, storing, 83
numeric data, 82
numeric fields, format statements for, 84

O

omitting, fields from *AR* prompting, 68
options, available via *AN*, 25
∨ *or*, use in constraints, 19
order of field prompts, with *AR*, 68
order of fields, in reports, 20 33
organization of MABRA systems, 16
output format, of data, 83
overwriting, a MABRA system, 79
owner, of a MABRA system, 81

P

page depth, of reports, 39
pagebreak star (*), 38
( ) *parentheses*, use in constraints, 20
use in printing code descriptions, 20 37
PAUSE, *GR* special option, 39
pausing, after page of output is printed, 39 57
percentages, produced with *AN*, 26
PF command, 53 56
positioning paper, for output, 17 39
pre-answering, 48 67
PREVIOUS, constraint, 21 44
print depth, 39
print fields, for reports, 20 33 37
print file, 75
print specifications, 24
printing, code descriptions in reports, 20 37
forms, 53 56
large number of records, 24
mailing labels, 53
record data on forms, 53
records, see *listing, records*
processing, update requests, 76
prompts for input, 14
pseudo fields, 36 50 81 87 91 94
created using *LS*, 44
purging, deleted records from the master file, 106

Q

□*FMT*, see *format statements*

R

RA command, 105
range points, for *AN*, 26 27
RB command, 106
RC command, 77 109

REAL, data type, 82
real fields, 81 91 94
reblocking, the master file, 106
recalling, stored responses to MABRA prompts, 47
record, definition of, 2
record access, 97 98
record details, including in a report, 34 38
record numbers, 22 107
record size, 108
redefining, stored answer-aheads, 48
reducing, data entry time, 87 92
removing, access to a MABRA system, 105
  blank rows from a form, 55 61
  deleted records from the master file, 106
  extra blanks from forms, 55 61
  update requests, 77
RENAME command, 107
renaming, a MABRA system, 107
report formatting, 33
report title, 35
report width, 34 40
reporting, relationships between fields, 25
reports, with matrix or template format, 53
resizing, a MABRA system, 106
retained pseudo fields, 88 94
retrieving, records, see *constraints*
returning, to APL immediate execution mode, 15
  to system mode, 15
reviewing, update requests, see *listing, update requests*
revoking, access, see *removing, access*
rows, of a crosstabulation analysis, 28 32
  of a form, 54
RS command, 106

S

SA command, 98 105
saving, data entry time, 87 92
  responses to MABRA prompts, 46
searching, records, see *constraints*
searching improvements, 91
security, see *access to a M system*
  see *locking, a M system*
; *semi-colon*, use in pre-answering, 48 67
session pseudo fields, 36 44 88 94
SETCREATOR, 17
setting, access to a MABRA system, 98
  system size, 78 80 106
setting up forms, 53
SF command, 53
sharable commands, 98
sharing, access to a MABRA system, 97
sign-on password, storing, 100
size, of a MABRA system, 78 80 106
/ *slash*, sortbreak slash, 35 38
snapshot, of a MABRA system, 76
sort fields, 21 33 38
sortbreak slash (/), use of, 35 38
sorting records, for reports, 21 33 38
SORTREQ, 25 53
source system, 78
SPACING, *GR* special option, 40
spacing, between lines of a report, 40
special options, for *GR*, 39
specifying, alternate column headings, 20 35 37
  forms, 53
  range points for data, 27
  system size, 78 80 106
stacking input, see *pre-answering*
START, 14
status, of a batch print job, 25 53
  of a MABRA system, 108
storage space, used by a MABRA system, 109
stored answer-aheads, 46
storing, descriptions for codes, 95
  responses to MABRA prompts, 46
  sign-on password, 100
STOP, 15
string searching, 20
subtotalling, for reports, 35
suppressing, field prompts, 68
system mode, 14
  command prompt, 14
  returning to, 15
system name, 80
  changing, 107
system number, 17 80
system size, 78 80
  changing, 106

T

target system, 78
terminating, commands, 15
  stored answer-aheads, 47
text items, on a form, 60
3-up labels, 54
~ *tilde*, use in constraints, 22
  use in specifying print fields, 20

TITLE, *GR* special option, 40
total fields, 33 36 38
totalling, data with *AN*, 25 29 31
  data with *GR*, 33 36 39
  data with *LS*, 43 46
trimming, of extra blanks on a form, 55 61
truncation of field values, 66
TYPE, field attribute, 94
types of data, 82 94
typing, more than one response on a line, 48 67

U

UP command, 76
  giving access to, 100
update file, 109
update requests, 68
  listing, 73
  removing, 77
  withdrawing, 75
update statistics, displaying, 109
update tasks, 76
  allowing other users to initiate, 100
updating, a MABRA system, 68 76
  records, 68 76
USDATES, state setting, 16
utilities, MABRA, 4
utility workspace, 4

V

validation, of input data, 66
variations, of fields, 21

W

width, of code descriptions, 92 96
  of fields, 83 94
  of reports, 34 40
WIDTH, field attribute, 94
WIDTH, *GR* special option, 40
withdrawing, an update request, 75
workspace, MABRA, 14
WU command, 75

Pub. Code: 0002 8310 E2

ISBN 0 86493 083 6